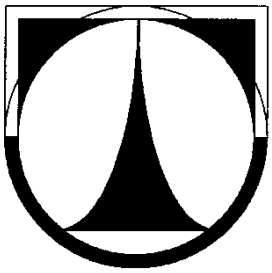
TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA MECHATRONIKY, INFORMATIKY A MEZIOBOROVÝCH STUDIÍ

Ústav nových technologií a aplikované informatiky



IMPLEMENTACE IPV6 DO DOMÁCÍCH SÍTÍ

IPV6 IMPLEMENTATION FOR HOME NETWORKS

BAKALÁŘSKÁ PRÁCE

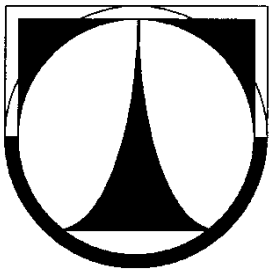
Vít Dittrich

Květen 2011

TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA MECHATRONIKY, INFORMATIKY A MEZIOBOROVÝCH STUDIÍ

Ústav nových technologií a aplikované informatiky



Studijní program

B2612 Elektrotechnika a informatika

Obor Informatika a Logistika

IMPLEMENTACE IPV6 DO DOMÁCÍCH SÍTÍ

IPV6 IMPLEMENTATION FOR HOME NETWORKS

Bakalářská práce

Vít Dittrich

Vedoucí bakalářské práce: doc. RNDr. Pavel Satrapa, Ph.D.

Počet stran: 47

Počet obrázků: 13

Počet tabulek: 1

Počet příloh: 0

Květen 2011

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2010/2011**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Vít DITTRICH**
Osobní číslo:	**M07000092**
Studijní program:	**B2612 Elektrotechnika a informatika**
Studijní obor:	**Informatika a logistika**
Název tématu:	**Implementace IPv6 do domácích sítí**
Zadávající katedra:	**Ústav nových technologií a aplikované informatiky**

Zásady pro vypracování:

1) Seznamte se s problematikou IPv6 a domácích směrovačů Ubiquiti na bázi OS Linux.
2) Navrhněte úpravu firmware pro tyto domácí směrovače, která by umožnila zavedení IPv6 do domácí sítě.
3) Poskytované funkce by měly zahrnovat směrování IPv6, podporu pro bezstavovou automatickou konfiguraci, možnost vytváření tunelů, a to včetně odpovídajících prvků GUI pro správu zařízení.
4) Navržené změny implementujte a otestujte.

Rozsah grafických prací: **dle potřeby**
Rozsah pracovní zprávy: **cca 40 stran**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] **SATRAPA, Pavel. IPv6 : internetový protokol IPv6. Praha : CZ.NIC, 2008. 357 s. ISBN 978-80-904248-0-7.**
[2] **SCHRODER, Carla. Linux : kuchařka administrátora sítě. Vyd. 1. Brno : Computer Press, 2009. 596 s. ISBN 978-80-251-2407-9.**
[3] **OpenWrt : Documentation [online]. 2010 [cit. 2010-10-13]. Dostupné z WWW: http://wiki.openwrt.org/doc/start.**
[4] **Ubiquiti Networks. Ubiquiti : Ubiquiti Wiki [online]. 2010 [cit. 2010-10-13]. Dostupné z WWW: http://www.ubnt.com/wiki/.**

Vedoucí bakalářské práce: **doc. RNDr. Pavel Satrapa, Ph.D.**
Ústav nových technologií a aplikované informatiky

Datum zadání bakalářské práce: **15. října 2010**
Termín odevzdání bakalářské práce: **20. května 2011**

V. Kopecký

prof. Ing. Václav Kopecký, CSc.
děkan

L.S.

prof. Dr. Ing. Jiří Maryška, CSc.
vedoucí ústavu

V Liberci dne 15. října 2010

Téma

Implementace IPv6 do domácích sítí

Anotace

Bakalářská práce se zabývá obecnou problematikou implementace protokolu IPv6 do domácích směrovačů. Pozornost je věnována příčině přechodu na nový protokol a stavu implementace v nejprodávanějších domácích směrovačích v České republice. Dále je popsán návrh implementace pro domácí směrovače firmy Ubiquiti Networks, Inc a následně jsou tyto změny aplikovány do upraveného firmware.

Cílem této práce je implementace navržených změn do funkčního firmware AirOS 5 tak, aby bylo umožněno provádět konfiguraci parametrů IPv6 nastavení pomocí grafického webového rozhraní zařízení.

Klíčová slova: IPv6, domácí sítě, AirOS, implementace.

Theme

IPv6 implementation for home networks

Annotation

The thesis deals with general issues of implementing protocol IPv6 in home routers. Focus is aimed at the occasion of change over to new protocol and implementation status in best selling home routers in the Czech Republic. Further on, the concept of implementing home routers from Ubiquiti Networks, Inc is described, followed by application of these changes to adapted firmware.

The aim of this paper is implementing designed changes to functional firmware AirOS 5 so that it's possible to make configuration alternations of parameters IPv6 facilitated by graphical web-interface of the device.

Key words: IPv6, home networks, AirOS, implementation.

Zpracovatel: TU v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií

Ústav nových technologií a aplikované informatiky

Dokončeno: 2011

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce.

V Liberci dne 20.5.2011 ……

podpis

Poděkování

Na tomto místě bych rád poděkoval vedoucímu bakalářské práce doc. RNDr. Pavlu Satrapovi, Ph.D. za odbornou pomoc, rady a konzultace.

Dále bych rád poděkoval především rodičům a všem mým blízkým, kteří mě po celou dobu studia na FM TUL podporovali a byli mi po dobu studia pevnou oporou.

Obsah

Přehled použitých zkratek, symbolů 9

1. Úvod 12

1.1. Cíle bakalářské práce 13

2. Historie protokolu IP 14

2.1. Historie protokolu IPv4 14

2.2. Historie protokolu IPv6 16

3. IPv6 v domácích směrovačích 19

3.1. Domácí směrovače obecně 19

3.2. Podpora IPv6 v domácích směrovačích 20

3.3. Domácí směrovače společnosti Ubiquiti Networks, Inc. 22

4. Návrh úpravy firmware 23

4.1. Výchozí stav 23

4.2. Konfigurace a diagnostika 24

4.3. Automatické konfigurace 25

4.4. Vytváření tunelu 26

4.5. Zabezpečení 27

5. Implementace úprav 29

5.1. AirOS V 29

5.2. Přístup skrze IPv6 29

5.3. Implementace 6to4 32

5.4. Automatická bezstavová konfigurace 33

5.5. Firewall 34

5.6. Ostatní nástroje webového rozhraní 37

5.7. Aplikace zvolené konfigurace 40

6. Testování 41

7. Závěr 45

Seznam použité literatury 46

Přehled použitých zkratek, symbolů

IP	Internet Protocol	
TCP	Transmission Control Protocol	
IPv4	Internet Protocol version 4	Internetový protokol verze 4
IPv6	Internet Protocol version 6	Internetový protokol verze 6
ITU	Internet Telecommunications Union	Mezinárodní telekomunikační unie
RFC	Request For Comments	označení řady standardů a dalších dokumentů popisujících systémy, Internetové protokoly apod.
IETF	Internet Engineering Task Force	Technická komise Internetu vyvíjející internetové standardy
IPng	Internet Protocol new generation	Internetový protokol nové generace
kbps	kilobit per second	kilobitů za sekundu
ISP	Internet Service Provider	Poskytovatel internetových služeb
TTL	Time To Live	Životnost paketu
OS	Operating System	Operační systém
GUI	Graphical User Interface	Grafické uživatelské rozhraní
SDK	Software Development Kit	Nástroj pro vývoj software
CIDR	Classless Inter-Domain Routing	Beztřídní směrování
IANA	Internet Assigned Numbers Authority	Organizace dohlížející na celosvětové přidělování IP adres, správu kořenových DNS, registr protokolů a další
DNS	Domain Name System	Systém doménových jmen
RIR	Regional Internet Registry	Regionální internetový registrátor
RIPE NCC	Réseaux IP Européens	Evropský regionální internetový registrátor
LIR	Local Internet Registry	Lokální internetový registrátor
NAT	Network Address Translation	Překlad síťových adres

SOHO	Small Office/Home Office	Malá kancelář/domácí kancelář
ISO/OSI	International Organization for Standardization /Open Systems Interconnection	Referenční komunikační model mezinárodní organizace pro stanadizaci
LAN	Local Area Network	Lokální (místní) síť
WAN	Wide Area Network	Síť pokrývající rozsáhle geografické území – například internet
CLI	Command-Line Interface	Příkazová řádka
TDMA	Time Division Multiple Access	Sdílení kapacity střídáním časových intervalů
GNU GPL	GNU General Public License	Všeobecná veřejná licence GNU
PHP	Hypertext Preprocessor	Hypertextový preprocesor
MIPS	Microprocessor without Interlocked Pipeline Stages	Architektura mikroprocesoru
MTU	Maximum Transmission Unit	Označení pro maximální velikost IP Datagramu
PPPoE	Point-to-Point Protocol over Ethernet	Komunikační protokol linkové vrstvy pro přímé propojení dvou uzlů skrze ethernetovou síť umožnující autentizaci, šifrování a kompresi dat.
DMZ	DeMilitarized zone	část sítě oddělená od ostatních zařízení
MAC address	Media Access Control address	Jedinečný identifikátor síťového zařízení nebo rozhraní
PING	Packet InerNet Groper	Program ověřující funkčnost spojení mezi dvěma síťovými uzly
DHCP	Dynamic Host Configuration Protocol	Protokol pro automatické nastavování síťových parametrů koncovým zařízením
DUID	DHCP Unique Identifier	Unikátní identifikátor pro DHCP

ARP	Address Resolution Protocol	Protokol k získávání MAC adres adresy sousedního zařízení
RA	Router Advertisement	Ohlašování směrovače

1. Úvod

Podíváme-li se do historie vývoje počítačových sítí a především pak internetu, zjistíme, že poptávka po spojení s celosvětovou sítí roste. Stejně tak se zlevňováním osobních počítačů a tím i větší dostupností pro širší okruh osob i firem, implementací podpory do mobilních a nejrůznějších domácích zařízení a spotřebičů roste i počet zařízení k sítím připojených. S touto potřebou rostla rapidně i potřeba IP adres pro všechna tato zařízení. Struktura počítačových sítí a tím pádem i internetu se však odrážela od protokolu IPv4, jenž má své kořeny na konci sedmdesátých a začátku osmdesátých let.

V té době však nikdo nepočítal s tím, že by zhruba 4 miliardy adres pro lidstvo nestačily, a proto se i přidělování těchto bloků řešilo poměrně nerozvážnou formou. Ačkoliv se postupnými změnami přidělování adresních bloků zamezilo dalšímu výraznějšímu plýtvání s adresami, které se ovšem netýkalo již přidělených adresních bloků, kompletní vyčerpání omezeného množství adres bylo již značně urychleno a ani následné zpomalování dalšími metodami vzhledem ke stále rostoucí poptávce dostatečně nezbrzdilo tempo, jakým IPv4 adresy docházely. Vzhledem k omezenému množství IPv4 adres bylo jisté, že tyto adresy budou dříve či později vyčerpány, a proto bude třeba vytvořit protokol, který by IPv4 definitivně nahradil.

V první polovině devadesátých let tak začala vznikat definice nového protokolu, který měl IPv4 nahradit. Nový protokol si však nekladl za cíl pouhé zvětšení adresního prostoru, který byl a dodnes je hlavním motorem celého přechodového procesu, avšak i další vylepšení, která při návrhu IPv4 nebyla brána v potaz a jejichž potřebu ukázala především další léta reálného provozu. Mezi takovéto mechanismy patří například podpora služeb se zaručenou kvalitou, bezpečnostní mechanismy, design odpovídající vysokorychlostním sítím, podpora mobilních sítí, automatická konfigurace a další.

Přestože byl nový protokol navrhnut tak, aby přechod ze staršího protokolu IPv4 byl co nejhladší, není s protokolem IPv4 zpětně kompatibilní, avšak tyto protokoly mohou paralelně koexistovat. Existují také nejrůznější druhy tunelů, které dokážou skrze fungující infrastrukturu IPv4 sítě zajistit IPv6 konektivitu.

Ačkoliv se první implementace protokolu IPv6 objevila již v průběhu roku 1998, dosud zaujímá průměrný datový přenos na novějším protokolu - pouze řádů desetin procenta

v celkovém objemu všech datových přenosů v síti internet. Tento fakt vychází především z toho, že řada sítí vůbec nový protokol nepodporuje a v případě, že je podporován, často nový protokol nepodporují právě domácí směrovače koncových uživatelů sítě internet. To je pak dále umocňováno nedostatkem obsahu v IPv6 sítích. Z důvodu nedostatku obsahu v IPv6 není nastolen dostatečný tlak na další implementaci IPv6 do sítí, což má za následek nedostatek uživatelů v globální IPv6 síti. A právě nedostatek uživatelů bývá častým důvodem pro mizivou podporu IPv6 ze strany poskytovatelů internetového obsahu, což tvoří jakýsi začarovaný kruh, který se zatím daří rozbít pouze pozvolna. Jednou z možností, jak tento kruh rozbít, je implementace IPv6 do domácích sítí a tím i zvýšení počtu uživatelů internetu využívajících protokolu IPv6.

1.1. Cíle bakalářské práce

V této bakalářské práci budu řešit návrh a úpravu firmware AirOS V založeném na bázi OS Linux pro zařízení firmy Ubiquiti Networks, Inc. tak, aby ve výsledném firmware byla zahrnuta podpora IPv6 včetně odpovídajících prvků v GUI.

V práci se budu nejprve zabývat problematikou IPv6 a domácích směrovačů, teoretickým návrhem podporovaných funkcí – především podporou pro bezstavovou automatickou konfiguraci a vytvářením tunelů, následně pak jejich praktickou implementací do aktuální verze firmware pomocí dodávaného vývojového nástroje (SDK). Výsledné úpravy v praxi otestuji.

2. Historie protokolu IP

2.1. Historie protokolu IPv4

Historie protokolu IP sahá do roku 1974, kdy dva američtí informatici Vinton Gray Cerf a Robert Elliot Kahn vydávají společnou publikaci s názvem *"A Protocol for Packet Network Intercommunication"*. V této publikaci se pak především zabývají tématem, jak vytvořit takovou počítačovou síť, do které by bylo možné připojit takřka neomezené množství počítačů a aby kterákoliv její část mohla fungovat samostatně, bez centrálních řídících prvků. Výsledkem jejich dlouhodobého snažení byl základ protokolu IP používaný v rámci počítačových sítí a internetu v jeho dalších revizích dodnes.

Hlavní inovací oproti předchozím mechanismům je rozdělení souboru dat na takzvané pakety opatřené hlavičkou, která obsahuje především adresu odesílatele a adresu příjemce daného paketu. Každý z těchto paketů je pak schopen počítačovou sítí putovat zcela samostatně a mnohdy i zcela odlišnou cestou.

Čtvrtou a nejrozšířenější revizí internetového protokolu je takzvaný IPv4. Struktura této revize IP je popsána v RFC 791[5] ze září roku 1981. IPv4 popisuje způsoby, jak spolu mají jednotlivé počítače propojené skrze počítačovou síť komunikovat. IPv4 je datově orientovaný protokol síťové vrstvy (modelu TCP/IP), který je používán v sítích s přepojováním paketů. Jde o protokol přepravující data bez jakékoliv záruky doručení dat, zachování správného pořadí nebo vzniku duplicitních paketů. Zajištění těchto vlastností přenosu dat je ponecháno na vyšší - transportní vrstvě v architektuře TCP/IP.

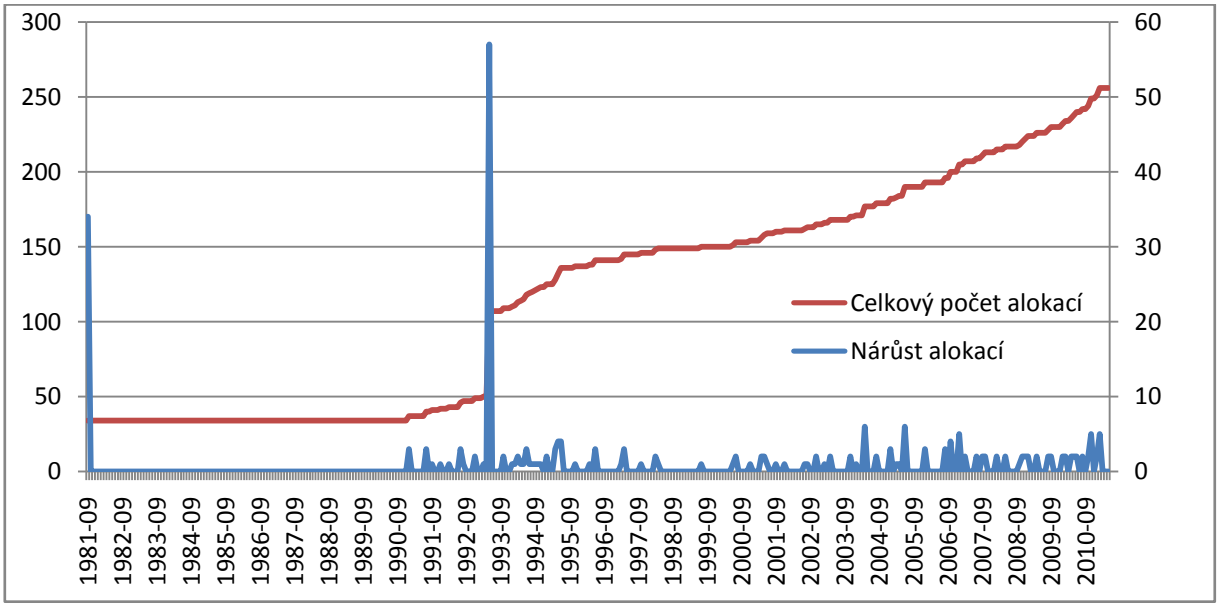
Jak již bylo zmíněno, každá hlavička paketu obsahuje zdrojovou a cílovou adresu. Tato IP adresa je v síti jednoznačné 32 bitové číslo zapisované pomocí čtveřice bytů oddělených tečkami (například 127.0.0.1 – binárně pak 01111111.00000000.00000000.00000001). Z toho vyplývá, že IPv4 disponuje teoreticky až 2^{32} (4 294 967 296) adresami. Tyto adresy se nacházejí v takzvaných podsítích - rozsazích. V každém rozsahu je však zapotřebí adresa sítě (network address) jednoznačně určující rozsah a broadcast adresa, které slouží k rozeslání dat všem adresám v daném rozsahu. Každý rozsah je tak „menší“ o dvě adresy.

V počátcích počítačových sítí byly adresy rozděleny do takzvaných tříd. Tyto třídy byly tři a byly označeny písmeny A, B a C. Síťové adresy třídy C byly určeny pro nejmenší sítě (co do počtu uzlů) a počítají jen s 256 (respektive 254) uzly, takže jedna takzvaná síťová adresa třídy C obnáší celkem 256 konkrétních IP adres. V případě, že takováto síť přerostla velikost 254

uzlů – byť jen o málo – adresní prostor velikosti C již síti nestačil. V řadě případů pak v minulosti dostala takováto síť adresní prostor velikosti B, který počítá již s 65 536 (respektive s 65 534) uzly. Takovýto blok adres pak nenávratně tato síť odčerpala ze společného IP prostoru. Pokud se tedy jednalo o síť kupříkladu s 500 uzly, dalších zhruba 65 000 adres zůstalo nevyužito, a to bez možnosti využití jinde právě kvůli specifické síťové části adresy. Stejný problém nastává u adres třídy A, jež počítala s více jak 16 miliony uzlů.

Vzhledem k tomu, že takovýmto způsobem se již brzy v počátcích přišlo na přílišné plýtvání adresami, adresy třídy A se takřka přestaly vydávat. S adresami třídy B to postupem času bylo stejné, a proto se například pro síť o 1000 uzlech přidělilo několik adres třídy C. Toto řešení se však na straně druhé odrazilo v nepříjemném růstu směrovacích tabulek, přesto nebylo stále dostatečně granulární pro stávající potřeby.

Problémy rychlého a zbytečného přidělování velkých adresních bloků se ve velké míře projevily v první polovině 90. let 20. století, jak znázorňuje i obrázek 2.1.1. V této době již bylo nutné změnit mechanismy přidělování rozsahů tak, aby bylo možné adresy přidělovat s ještě větší granularitou. Toho lze dosáhnout připojením externího údaje o pomyslné dělicí čáře mezi síťovou částí adresy a částí patřící danému uzlu. Tento přístup pod názvem Classless Inter-Domain Routing (CIDR) tak zrušil třídy a dovolil agregaci sousedních rozsahů.

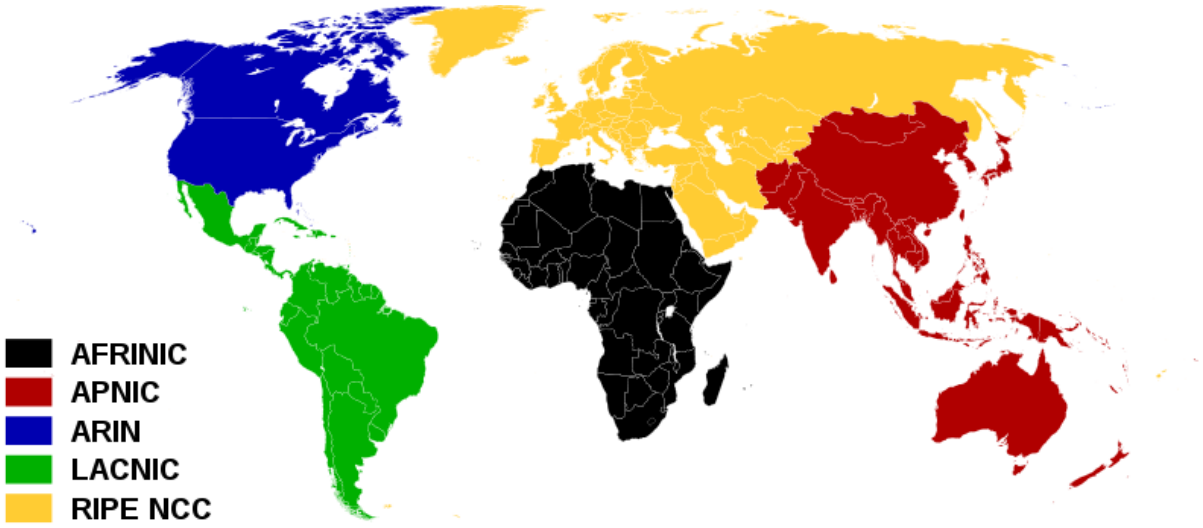


Obr. 2.1.1: Spotřeba IPv4 adres (zdroj http://www.iana.org/numbers/)

Příslušnost k dané síti se určuje prefixem, kdy všechna rozhraní v jedné síti mají stejný začátek adresy. Prefix je stejně jako u IP adresy označován 32 bity. Celkově tedy existuje 33 typů podsítí (0-32), které se liší svou velikostí a kde prefix /0 obsahuje všechny myslitelné IP

rozsahy. Maska sítě zapsaná v binárním tvaru má zleva samé jedničky až do místa, kde končí číslo sítě a dále pak samé nuly na místě pro část síťového rozhraní. Podle počtu jedniček se pak tomuto číslu říká prefix. Prefix /24 tedy můžeme binárně zapsat jako 11111111.11111111.11111111.00000000 nebo dekadicky jako 255.255.255.0 a to vše se stejným významem.

Přidělováním IPv4 se na různých úrovních zabývají různé organizace. Na globální úrovni to je organizace IANA, která přidělovala rozsahy zpočátku přímo konkrétním zákazníkům a později pouze regionálním registrátorům. Těch je ve světě pouze 5, což je znázorněno na obrázku 2.1.2. Přes všechny další mechanismy, které zpomalily vyčerpání IPv4 adresního prostoru jako například NAT nebo další nejrůznější zpřísnění přidělování nových adres lokálním registrátorům, poslední adresy s prefixem /8 byly na globální úrovni rozděleny 3. února 2011 na konferenci v Miami, USA.



Obr. 2.1.2: Regionální registrátoři (RIR)

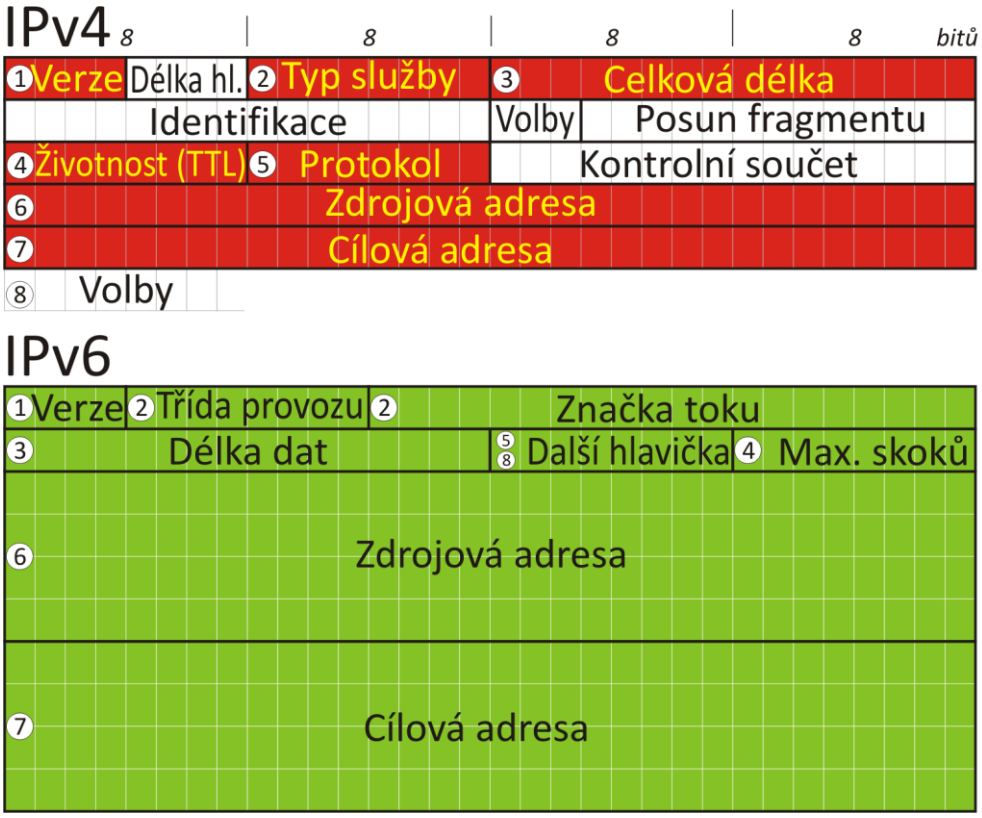
2.2. Historie protokolu IPv6

Počátkem devadesátých let 20. století již bylo zřejmé, že přes veškerá opatření jako CIDR a NAT nebude počet IPv4 adres pro stále narůstající počet zařízení stačit. Koncem roku 1992 tedy přišlo konsorcium IETF s výzvou prostřednictvím RFC 1550[6] pro podání návrhu Internetového protokolu nové generace - IPng později přejmenovaného na IPv6. Základní požadavky byly následně vyhodnoceny v RFC 1752[7] a byly ustanoveny pracovní skupiny.

Koncem roku 1995 byla vydána první generace definic nového protokolu počínaje RFC 1883[8], a to pod názvem, jak jej známe dodnes. Již v tomto RFC jsou hlavním tématem změny

oproti IPv4 v těchto oblastech: schopnost rozšíření adresního prostoru, zjednodušení formátu hlavičky paketu, zlepšení podpory pro rozšíření a volitelné možnosti hlavičky, schopnost označení datového toku a jako poslední oblast rozšíření o podporu autentifikace a zabezpečení dat. Nový protokol tedy neměl pouze zvětšit adresní prostor a zbytek jen převzít z IPv4, ale měl již v samotném standardu obsahovat to, co v době návrhu v IPv4 chybělo a co bylo mnohdy velmi složitě dodělávano a bylo tak v IPv4 mnohdy palčivým tématem.

V době vzniku tohoto RFC počítali tvůrci s poměrně rychlým přechodem z IPv4 na IPv6, a to v horizontu zhruba deseti let. Klíčovým přibrzděním se však staly již zmiňované mechanismy – především směrování na základě síťové masky (CIDR) a NAT, které prodloužily životnost adresního prostoru IPv4. To v praxi vedlo k velkému zbrzdění snahy výrobců hardware i software, kteří nebyli pod takovým tlakem na zavádění nového protokolu do svých produktů.



Obr. 2.2.1 Porovnání hlaviček IPv4 a IPv6[4]

Navzdory všem těmto problémům však už v průběhu listopadu roku 1996 byla v Linuxovém jádru 2.1.8 přidána experimentální vlastnost IPv6 a další systémy na sebe nenechaly dlouho čekat, avšak podporu bylo často nutné doinstalovat. Například v rodině operačního systému Windows určeného pro koncové uživatele bylo možné doinstalovat

podporu IPv6 až u Windows XP Service Pack 1, jehož vydání přišlo v roce 2002. Standardní součástí systému je však podpora IPv6 až od verze Windows Vista, který byl vydán v roce 2007.

Vzhledem k faktu, že nativní IPv6 konektivitu bylo v počátcích možné zajistit velmi zřídka, vznikaly v počátcích pouze oddělené ostrůvky IPv6 sítě. V tomto ohledu měla pomoci – především pro testování a experimentování s reálným provozem - experimentální síť s názvem 6bone. Tento projekt byl spuštěn v průběhu roku 1996. Tato síť propojovala ony oddělené ostrůvky IPv6 pomocí tunelů skrze již fungující IPv4 infrastrukturu. Ačkoliv se v průběhu fungování sítě 6bone část tunelů změnila na nativní spoje, převážná část infrastruktury zůstala až do konce fungování sítě právě v podobě tunelů. Vzhledem k faktu, že 1. 7. 1999 byly regionálním registrátorům (RIR) přiděleny první produkční IPv6 prefixy, které se v dalším období začaly přidělovat lokálním registrátorům a dále pak koncovým uživatelům, začala síť 6bone ztrácet svůj původní význam. Proto v roce 2004 vyšlo RFC 3701[9] oznamující postupné ukončení činnosti této experimentální sítě. 6. 6. 2006 přestaly být prefixy sítě 6bone dále směrovány, což v praxi ukončilo činnost této sítě.

Koncem roku 1998 přichází nová generace RFC počínaje RFC 2460[10], která nahrazuje starší již zmíněnou generaci. Tato generace se stává základním kamenem IPv6. Jedním z hlavních přínosů RFC 2460 je přesná definice formátu datagramu tak, jak jej známe dnes, který je zobrazen na obrázku 2.2.1.

Ačkoliv základní definice IPv6 protokolu existují již poměrně dlouho, konkrétní specifikace některých převážně podpůrných mechanismů se objevují nebo jsou aktualizovány dodnes. I to je jeden z důvodů, proč se nástup IPv6 začíná i v novém tisíciletí značně protahovat. Velkou úlohu při nenasazování IPv6 sehráli taktéž poskytovatelé internetového obsahu a ISP. Ti neposkytovali obsah a služby na IPv6, jelikož nebylo příliš mnoho uživatelů, kterým by tyto služby mohli poskytnout. A stejně tak nebylo mnoho uživatelů v IPv6 sítích, protože nebyl poskytován téměř žádný obsah.

Vzhedem k nečinnosti výrobců hardware, poskytovatelů internetu a internetového obsahu zakročily v řadě případů do rozšíření IPv6 vlády a vládní nařízení. V roce 2008 byl v Bruselu vydán akční plán na rožšíření Internetového protokolu verze 6 s názvem *ADVANCING THE INTERNET*[12], kde se mimo jiné plánuje zpřístupnění IPv6 sítě pro 25% uživatelů do roku 2010. V dubnu 2011 však nedosahuje průměr EU ani 10%. Česká vláda v roce 2009 přijala usnesení číslo 727[13], kde mimo jiné ukládá zajistit do 31.12.2010 přístup

k internetovým stránkám a veřejně dostupným službám eGovernmentu internetovým protokolem verze 4 i verze 6. Bohužel však ani v této době (duben 2011) není velká část těchto stránek a aplikací po IPv6 dostupná.

Velkým mezníkem v nárůstu uživatelů připojených na IPv6 by tak mohlo být postupné vyčerpání IPv4 adresního prostoru. Na globální úrovni sice IPv4 adresy již došly, než se však tento problém více reálně dotkne i koncových uživatelů, bude to pravděpodobně ještě nějakou dobu trvat. Záležet přitom bude na konkrétním regionu. Regionální registrátor APNIC 15.4.2011 začal přidělovat IPv4 adresy z posledního rozsahu[15] s prefixem /8. Lze tedy předpokládat, že vzhledem k vyčerpání IPv4 adres bude v tomto regionu první masivnější nasazení IPv6.

3. IPv6 v domácích směrovačích

3.1. Domácí směrovače obecně

Směrovače jsou obecně aktivní síťová zařízení, která přeposílají datagramy z jednoho rozhraní na jiné směrem k jejich cíli. Tento proces se nazývá směrování. Routování probíhá na třetí - síťové vrstvě (Layer 3 - L3) referenčního modelu ISO/OSI. Pro domácí směrovače se obecně vžilo označení jako SOHO routery.

Domácí směrovače jsou zařízení, která oddělují domácí síť LAN od sítě internetového poskytovatele. Většinou se v takovýchto domácích sítích používají neveřejné (privátní) IPv4 adresní rozsahy definované v RFC 1918[11]. Tyto rozsahy nejsou globálně směrovány a jednotlivé IPv4 adresy jsou unikátní pouze na úrovni lokální sítě. Z těchto důvodů pak musí domácí směrovač často zastávat ještě funkci NAT routeru, který přepisuje u průchozích paketů zdrojovou a/nebo cílovou IPv4 adresu v datagramu na lokální adresu. Často ještě může v datagramech přepisovat čísla TCP/UDP portů. Dále musí provést i změnu kontrolního součtu, aby změny byly brány v potaz a odpovídaly tak změněným hlavičkám. Směrovač si často dále musí uchovávat tabulku spojení tak, aby mohl ke správným paketům z internetu přiřadit cílovou adresu z lokální sítě. Všechna zařízení lokální sítě se tak skryjí za jedinou IPv4 adresu.

Takovéto řešení má především bezpečnostní výhody. Další výhodou je možnost připojit velké množství zařízení do sítě bez nutnosti přidělení dalších veřejných adres. Mezi hlavní

nevýhody však patří především možnost navázání spojení jen z vnitřní sítě směrem ven. Obráceně to – pokud není například nějaký port přímo směrován na konkrétní vnitřní IPv4 adresu - není možné, jelikož NAT router nerozpozná, které konkrétní IPv4 adrese dané spojení přiřadit. To může pro některé aplikace být i nepřekonatelný problém a nemusí pak v této konfiguraci sítě správně nebo dokonce vůbec fungovat.

3.2. Podpora IPv6 v domácích směrovačích

Tato část práce se zabývá stádiem implementace IPv6 do domácích bezdrátových směrovačů různých výrobců. Budu přitom vycházet z dat společnosti i4wifi a.s., která je jednou z největších společností prodávající bezdrátová zařízení na českém trhu.

Dle statistik i4wifi a.s. byla v roce 2010 nejprodávanější zařízení firmy Ubiquiti Networks, Inc., kterých bylo prodáno 45% z celkových více jak 700 000 kusů zboží. Zhruba 18% podíl na trhu zaujala firma MikroTik. Dále pak firmy TP-Link, CC&C a OvisLink. Další významnou firmou, která však není v sortimentu firmy i4wifi je firma D-Link.

Prvně zmiňovaná společnost vyrábí bezdrátové směrovače ve dvou kategoriích. Starší série pro normy 802.11a/b/g a novější s podporou normy 802.11a/g/n s technologií TDMA pojmenovanou jako AirMAX. Obě tyto série mají vlastní firmware pojmenovaný AirOS, který je založen na Linuxové distribuci OpenWRT. Pro starší výrobky se momentálně jedná o AirOS 3.5.1 a pro novější pak AirOS 5.3. Ani jeden z těchto firmware ale standardně nepodporuje IPv6. Přesto je možné pomocí dodávaného vývojového prostředí (SDK) tuto podporu doplnit. Výrobce nicméně v budoucnu slibuje podporu IPv6 v jím dodávaném firmware. Do obou sérií zařízení lze nahrát i vlastní firmware třetích stran, ať se již jedná o některou upravenou Linuxovou distribuci, nebo některé komerční firmware. Zde pak podpora IPv6 záleží na konkrétním firmware.

Druhou jmenovanou společností je firma MikroTik. Tato firma se zabývá vývojem a výrobou nejen bezdrátových směrovačů a bezdrátových karet, ale i čistě ethernetových směrovačů a v poslední době i takzvaných SMART switchů. Zařízení této lotyšské firmy disponují firmware pojmenovaným RouterOS založeným opět na operačním systému Linux, který se konfiguruje buď pomocí konzole (CLI), proprietární aplikace Winbox nebo nově i pomocí webového rozhraní. RouterOS lze také provozovat na PC platformách

s architekturou procesoru x86. RouterOS v nynější stabilní verzi 5.0 plně podporuje IPv6. Podpora IPv6 je zde již od verze 3.0 a je postupně vylepšována.

Další výrobky prodávané v České republice jsou výrobky firmy TP-Link. Bohužel originální firmware prozatím nepřidává podporu IPv6. Do některých konkrétních výrobků často lze nahrát jiný firmware, který může IPv6 podporovat. Jedná se tak například o firmware založený na Linuxové distribuci OpenWRT.

U výrobků firmy CC&C je situace různá dle typu zařízení. U některých je zmiňována transparentní podpora IPv6, ale tento termín není nikde dále rozveden. Žádná podpora u těchto zařízení ale nebyla zaznamenána. U jiného zařízení stejného výrobce je oficiálně deklarována plná podpora IPv6. Opětovně jsem však ve firmware nenarazil na jakýkoliv prvek, umožňující jakékoliv nastavení. Lze tedy předpokládat, že zařízení za určitých okolnostní IPv6 pakety alespoň propouští, což ostatně dělá za určitých okolností i většina ostatních výrobků jiných firem.

Zařízení firmy OvisLink taktéž v originálním firmware IPv6 nepodporují. Stejně jako u některých předchozích výrobců je ale možné nahrát do zařízení jednu z upravených Linuxových distribucí, kde by mělo být možné nastavit podporu IPv6.

Poslední zmiňovanou značkou je firma D-Link. Ta u novějších výrobků deklaruje podporu IPv6 ve standardně dodávaném firmware. Bohužel z veřejně dostupných zdrojů nebylo možné zjistit, co všechno podpora IPv6 v podání firmy D-Link obsahuje.

Pro přehledné srovnání je v tabulce 3.1.1 shrnut stav podpory IPv6 ve firmware vybraných výrobců. Je zde patrné, že podpora IPv6 ve standardním firmware u řady produktů zcela chybí, což brání masovějšímu nasazení IPv6. U řady zařízení sice lze podporu získat změnou na firmware třetí strany, avšak takovýto zásah do zařízení často vyžaduje jistou odbornou znalost.

Výrobce	Podpora v originálním firmware	Lze doplnit v SDK	Lze zajistit s firmware třetí strany
Ubiquity Networks, Inc.	Není	Ano	Ano
MirkroTik	Plná	Podpora v originálním firmware	Ano
CC&C	Výrobce udává podporu	Nedodává SDK	Není známo
TP-Link	Ne	Nedodává SDK	Ano
Ovislink	Ne	Nedodává SDK	Ano
D-Link	Ano	Podpora v originálním firmware	Ano

Tabulka 3.1.1: Podpora IPv6 ve firmware

3.3. Domácí směrovače společnosti Ubiquiti Networks, Inc.

Společnost Ubiquiti Networks, Inc. dodává na český trh zařízení ze dvou hlavních sérií. V obou sériích je použit vlastní firmware založený na Linuxové distribuci OpenWRT s názvem AirOS. Ve starší sérii je nejnovější AirOS verze 3.6.1 a pro novější pak AirOS verze 5.3.

K oběma verzím je dodáváno vývojové prostředí, ve kterém je možné provést úpravy originálního firmware. Většina součástí systému je vydávána pod licenci GNU/GPL, avšak některé části - jako například ovladače bezdrátových karet nebo ethernetu, mají uzavřený zdrojový kód. Je však možné tyto části nahradit volně šiřitelným software.

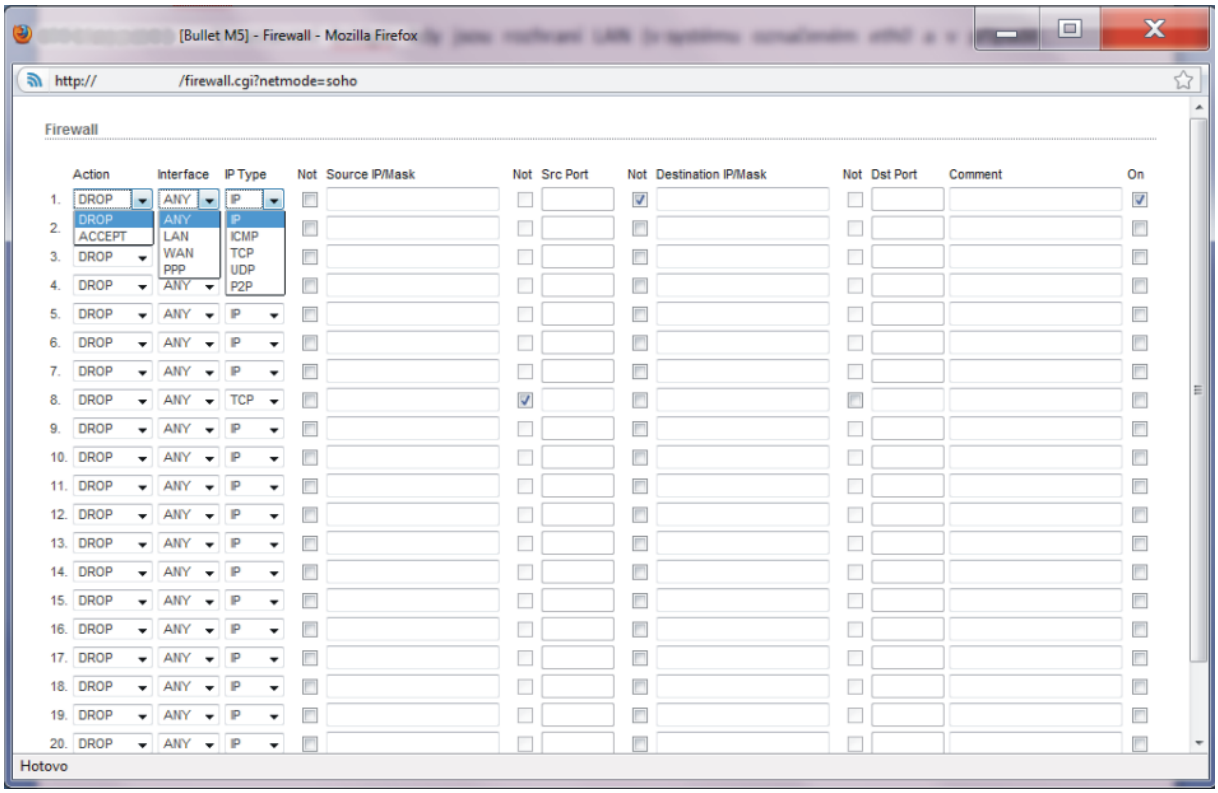
Novější série s podporou firmware AirOS 5 je postavena na chipsetu firmy Atheros s procesorem architektury MIPS. Dále tato zařízení obsahují 8MB paměti typu NAND a RAM velikosti 32MB. I z těchto důvodů musí být firmware pro tato zařízení velmi minimalistický a obsahovat pouze nejnutnější sady nástrojů. V originálním firmware se nedostatek místa a paměti odráží například na skriptovacím jazyku PHP, na kterém je postaveno webové GUI. Ve firmware je požito PHP verze 2 z roku 1996, ačkoliv nejnovější verze z ledna 2011 je již s pořadovým číslem 5.3.5.

4. Návrh úpravy firmware

4.1. Výchozí stav

V originálním firmware AirOS verze 5.3 ani v připravované betaverzi AirOS 5.5 není v jádře systému zahrnuta podpora IPv6. Navzdory tomu je ve webovém konfiguračním rozhraní velmi dobře propracovaná základní konfigurace zařízení a nastavení IPv4. Zařízení lze konfigurovat pomocí grafického webového rozhraní, nebo skrze příkazovou řádku, kterou lze vzdáleně zpřístupnit protokolem SSH. Co se týče síťového nastavení, uživatel si může vybrat ze tří síťových módů, a to bridge, router a SOHO router.

V nastavení bridge jsou rozhraní LAN (v systému označeném *eth0* a v případě víceportového zařízení pak ještě eth1) a WLAN (v systému označeném *ath0*) softwarově spojeny na druhé (linkové) vrstvě referenčního modelu ISO/OSI do rozhraní bridge (v systému označovaném jako *br0*). V tomto módu lze nastavit IP adresa pro rozhraní bridge – na výběr máme statickou IP adresu nebo adresu dynamicky přidělenou DHCP serverem.



Obr. 4.1.1 Originální IPv4 firewall

V nastavení router a SOHO router je možné nastavit IP adresy pro každé rozhraní zvlášť. Pro rozhraní lokální sítě zde lze nastavit IP adresu, umožnit překlad adres NAT, nastavit

přesměrování portů a zapnout DHCP server. U rozhraní určeného pro síť WAN je na výběr z nastavení statické adresy, dynamické adresy přidělené DHCP serverem nebo skrze PPPoE. V poslední řadě zde můžeme nastavit DMZ nebo změnit MAC adresu.

Pro všechny tři módy je možné nastavit IP aliasy, neboli konkrétnímu rozhraní přiřadit více IP adres a pro každé rozhraní nastavit velikost MTU. Ve všech módech zde také lze nastavit statické směrování a firewall. U statického směrování máme na výběr 20 položek, kdy nastavíme cílovou síť, masku, bránu a záznam můžeme opatřit komentářem. U firewallu je možné opět nakonfigurovat až 20 záznamů, jak je zobrazeno na obrázku 4.1.1. Mezi základní diagnostické nástroje originálního webového rozhraní patří především program ping, který umožňuje prověřit funkčnost spojení mezi dvěma síťovými uzly.

Vzhledem k propracovanosti originálního firmware by bylo vhodné u navrhovaných změn co nejvíce zachovat originální design a přizpůsobit funkčnost tak, aby tyto změny byly pro uživatele co nejintuitivnější.

4.2. Konfigurace a diagnostika

Zařízení podporující protokol IPv6, který se má v budoucnu stát primárním protokolem internetu a počítačových sítí, by mělo podporovat konfiguraci probíhající po tomto protokolu. Jak již bylo zmíněno, zařízení můžeme konfigurovat pomocí grafického webového rozhraní, nebo skrze příkazovou řádku zpřístupněnou pomocí protokolu SSH. Je proto potřeba zajistit nastavení programů obsluhujících zmíněné služby tak, aby byla možná konfigurace nejen pomocí IPv4.

Originální firmware taktéž obsahuje možnost zálohování konfigurace. Bylo by tedy vhodné upravit firmware takovým způsobem, aby i konfiguraci IPv6 bylo možné zálohovat v podobě textového souboru. Dále by měla být také přidána možnost obnovy takovéto konfigurace stejně jako je tomu v originálním firmware.

Je taktéž potřeba zajistit dostatečnou bezpečnost zařízení, aby přístupu ke konfiguračním nástrojům nemohlo být jednoduše zneužito například metodou brute force, nebo aby bylo toto riziko minimalizováno. Toho lze v prvé řadě dosáhnout omezením přístupu k webovému rozhraní nebo SSH pouze z předem definovaných adres nebo adresních rozsahů.

Je také potřeba provést takové modifikace, aby bylo možné diagnostikovat alespoň základní problémy. Toho lze velmi často docílit použitím programu ping. Program využívá zprávy „Echo Request“ (typ 8, výzva) a „Echo Reply“ (typ 0, odpověď) protokolu ICMP. Výzvy jsou odesílány na cílovou IP adresu a ve stanoveném limitu se očekává odpověď. V případě, že tento časový limit vyprší, je tento údaj považován za ztracený. Program průběžně vypisuje, které odpovědi již došly a s jakým zpožděním (latencí). Z toho lze diagnostikovat dostupnost daného uzlu, ztrátovost trasy a její zpoždění.

Dalším z diagnostických nástrojů je program traceroute. Ten slouží k analýze počítačové sítě, kdy vypisuje směrovače na cestě datagramů od zdroje až k cíli. Pomocí tohoto programu tak snadno zjistíme „cestu“, po které dané pakety putují. Toto je vhodné především, pokud máme v zařízení více síťových rozhraní a dokážeme tedy vyhodnotit, která trasa je pro daný cíl preferována.

4.3. Automatické konfigurace

Oproti IPv4 podporuje protokol IPv6 několik způsobů automatické konfigurace. Vzhledem k délce a relativní složitosti IPv6 adres je preferováno automatické nastavení stanic, aby byly kladeny co nejmenší nároky na jejich správce. Na výběr jsou dvě možnosti automatického nastavování – stavová a bezstavová konfigurace.

Stavová konfigurace je zajišťována serverem spravujícím konfigurační parametry, které pak server na požádání sděluje. Pro účely stavové konfigurace byl v IPv6 navržen protokol DHCPv6, který vychází z DHCP pro IPv4. Rozdíly oproti DHCP jsou však v některých oblastech značné. Především lze zmínit způsob identifikace klientů. Pro tento účel se nově zavádí unikátní DHCP identifikátor (DUID), který by měl být trvalý a neměl by záviset na klientově technickém vybavení. Je však definováno několik alternativ, jak operační systém identifikátor vygeneruje, a proto v každém operačním systému na stejném zařízení je většinou identifikátor jiný, a to například i při jeho přeinstalování. Oproti identifikaci pomocí MAC adresy v DHCP je tak identifikace koncových zařízení znatelně složitější a přiřazení fixní IPv6 adresy pak většinou prakticky nemožné.

Jednou z dalších velkých změn je u DHCPv6 delegace celých prefixů, a proto klienty DHCPv6 serverů nemusejí být nutně pouze koncové stanice, ale i směrovače. Tak jako u IPv4 dostal klient přidělenou IP adresu, zde dostane celou podsíť.

Další možností je automatická bezstavová konfigurace popsaná poprvé v RFC 1971[16] z roku 1996. Oproti DHCPv6 je automatická bezstavová konfigurace novinkou, nevyžaduje žádné speciální servery a je založena na objevování sousedů (Neighbor Discovery). Tento mechanismus slouží ke zjišťování linkových adres sousedních zařízení ve stejné podsíti, nahrazuje ARP z IPv4 a dále přidává funkce související právě s bezstavovou konfigurací a směrováním.

V principu každý směrovač v síti v náhodném intervalu rozesílá do sítí, ke kterým je připojen takzvané ohlášení směrovače – Router Advertisement (RA). Tato ohlášení může směrovač zaslat i na žádost zařízení. Ohlášení obsahují potřebné síťové parametry - především prefix dané podsítě a implicitní bránu. V původním návrhu automatické bezstavové konfigurace však v základních parametrech nebyla obsažena adresa DNS serveru a bylo proto nutné tuto adresu koncovým zařízením sdělovat jiným mechanismem - povětšinou pomocí bezstavového DHCPv6 nebo zajišťování DNS překladů po IPv4.

V září roku 2007 bylo vydáno RFC 5006[17] s experimentálním statusem, které zavádí oznamování DNS serverů jako volitelnou možnost. V listopadu roku 2010 bylo toto RFC nahrazeno novým RFC 6106[18], které přechází z experimentálního charakteru a navíc přidává možnost oznamování seznamu přípon doménových jmen. Bohužel kvůli krátké době od vydání tohoto RFC ještě není k dispozici plně funkční implementace.

Vzhledem k zadání práce a k nevýhodám spojeným s unikátním identifikátorem DHCPv6 by měla implementace zahrnovat nástroje pro bezstavovou automatickou konfiguraci - respektive v případě směrovače ohlašování prefixů a v případě síťového mostu pak možnost automaticky adresu získat.

4.4. Vytváření tunelu

Protože je v dnešní době poměrně nízká rozšířenost nativní IPv6 konektivity, měl by směrovač disponovat mechanismy, které umožňují připojit koncovou síť skrze fungující IPv4 infrastrukturu k IPv6 síti. Jelikož má být směrovač obecně použit pro přístup jiného zařízení

k počítačové síti a těchto zařízení může být více, je pravděpodobně nejjednodušší možností použít mechanismu 6to4. Pro účely tunelového spojení s IPv6 světem sice existují i jiné mechanismy, jako například 6over4, ISATAP nebo Teredo, avšak ty jsou určeny pro koncová zařízení a nikoliv pro celou podsíť.

Mechanismus 6to4 byl poprvé pospán v RFC 3056[19] z února 2001 a jeho oficiálním názvem je *Connection of IPv6 domains via IPv4 Clouds*. 6to4 se řadí mezi automatické tunely, kdy na základě jedné veřejné IPv4 adresy vygeneruje IPv6 prefix pro adresování celé koncové sítě a postará se o automatické tunelování datagramů mezi ní a zbytkem IPv6 internetu. Je tedy vyžadována pouze jedna veřejná IPv4 adresa, která musí mít spojení s IPv4 internetem.

IPv6 prefix pro danou podsíť se vygeneruje na základě této veřejné IPv4 adresy, kdy prvních 16 bitů obsahuje hexadecimální hodnotu 2002 a dalších 32 bitů do prefixu /48 se odvodí z IPv4 adresy. Z adresy 1.1.1.1 tudíž získáme prefix 2002:0101:0101::/48. S tímto prefixem pak můžeme zacházet jako s jakýmkoliv jiným standardním IPv6 prefixem. Lze jej proto rozdělit do potřebného počtu podsítí a přiřazovat IPv6 adresy koncovým zařízením.

Ve chvíli, kdy začneme z naší sítě komunikovat s IPv6 světem, „zabalí“ směrovač naše IPv6 datagramy do IPv4 a pošle tato data k takzvaným zprostředkovatelům. Ti mají nativní přístup jak do IPv4 tak do IPv6 světa. Přijaté IPv4 datagramy tedy „rozbalí“ a dále pracují s běžnými IPv6 datagramy.

Při implementaci by měl být kladen důraz na nenáročnost nastavení. Webové rozhraní by tak mělo obsahovat pole pro zadání IPv4 adresy, která bude při konfiguraci automaticky zpracována. Dále by bylo vhodné přidat možnost zajištění delegace zvolené podsítě z 6to4 prefixu na některé funkční rozhraní.

4.5. Zabezpečení

Jednou z hlavních změn IPv6 oproti IPv4 je to, že každá globální IPv6 adresa by měla být dosažitelná z celé IPv6 sítě. Ve své podstatě je tudíž každá globální IPv6 adresa veřejná. Z toho plyne jisté bezpečnostní riziko ať již samotného útoku proti směrovači, tak i proti koncovému zařízení. Vzhledem k zvětšujícímu se počtu uživatelů s připojením do IPv6 světa lze předpokládat, že útoků po IPv6 infrastruktuře bude přibývat.

Zabezpečení samotného směrovače bylo zmíněno v kapitole 4.2. Toto zabezpečení se však netýkalo připojených zařízení za směrovačem. V IPv4 sítích většinou pro základní zabezpečení postačoval NAT, který dovolil navázání spojení pouze z vnitřní sítě směrem do internetu. V opačném směru je možné navázat spojení za velmi specifických podmínek a za pomoci specifických nástrojů, a to většinou i na straně samotného NAT směrovače. V IPv6 nicméně standardně takováto možnost není a všechen provoz je navazován přímo, což bylo i jedním ze základních požadavků na samotný protokol. Za těchto podmínek se ale méně zabezpečená zařízení často stávají snadným cílem pro některý druh útoku. Dále je také možnost šíření počítačových virů z již infikovaných stanic do vnitřní sítě.

Řešením tohoto problému tak může být zakázání navazování příchozích spojení a povolení otevření spojení pouze směrem z vnitřní sítě dále do internetu. Tímto problémem se zabývá RFC 6092[20], ve kterém se mimo jiné poukazuje na fakt, že u většiny zařízení je využito pouhé minimum funkcí a zbytek zůstává v továrním nastavení. Zatímco u IPv4 sítí byl alespoň částečnou ochranou právě již zmiňovaný NAT, který jednoduše aplikovala řada správců nebo uživatelů, u IPv6 sítí je v takovýchto případech doporučeno v továrním nastavení zapnout alespoň základní ochranu koncových stanic.

Právě kvůli doporučením plynoucím z RFC 6092 by bylo vhodné implementovat možnost, kdy při zapnuté volbě bude možné navázat spojení pouze z vnitřní sítě stejně jako tomu bylo u IPv4 stanic za NAT směrovačem. Právě proto, že takovéto omezení je velmi striktní a s největší pravděpodobností bude celá řada aplikací, ve kterých je zapotřebí, aby byla navazována spojení právě z internetu – jako například web server – měl by k takovémuto zabezpečení existovat i plnohodnotný firewall, kde si můžeme nadefinovat výjimky nebo nastavit vlastní pravidla.

Na základě požadavku, kdy chceme pro základní ochranu koncových zařízení použít metodu, která zakáže navázání spojení směrem z internetu do vnitřní sítě, budeme muset využít stavový firewall. Ten nejen kontroluje IP datagramy, ale zároveň udržuje tabulku s již navázanými spojeními.

Pro implementaci do webového rozhraní by tedy bylo vhodné inspirovat se řešením firewallu pro IPv4 z originálního firmware a tuto verzi pak použít jako rozšířenou variantu pro vkládání vlastních pravidel.

5. Implementace úprav

5.1. AirOS V

AirOS verze 5 je firmware pro výrobky série AirMAX společnosti Ubiquiti Networks, Inc. Tento firmware je založen na Linuxové distribuci OpenWRT, která je učena právě pro tvorbu firmware do nejrůznějších embedded zařízení. OpenWRT disponuje balíčkovacím systémem IPKG, se kterým je možné jednoduše instalovat poměrně velkou část potřebného software a nastavovat parametry výsledného firmware jako typ filesystému, kompresi a další parametry výsledného obrazu systému. V případě, že požadovaný balíček není k dispozici, je možné jej vytvořit nebo software zkompilovat manuálně. Jakékoliv další soubory nebo potřebné obslužné programy, které nejsou zahrnuty v balíčku, lze zkopírovat do adresářové struktury výsledného firmware.

AirOS V je založen na starší verzi systému OpenWRT, a to konkrétně verzi s označením Kamikaze s Kernelem verze 2.6.15. Programátoři firmy Ubiqity přidali vlastní webové rozhraní pro konfiguraci zařízení. Dále pak ovladače bezdrátové karty a ethernetu, systém pro upgrade firmware a zálohu nastavení - to však již s uzavřeným zdrojovým kódem.

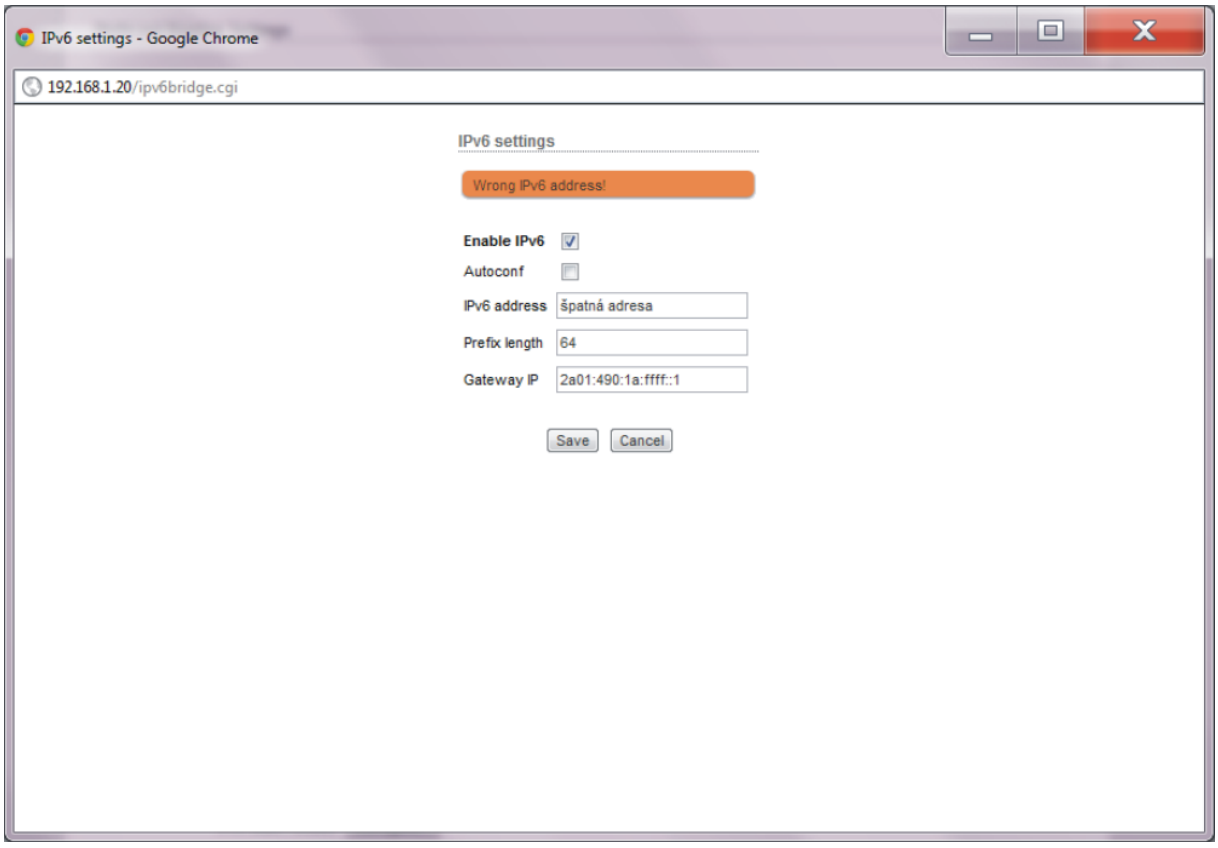
Tvorba firmware probíhá za pomoci připravených nástrojů na počítači s operačním systémem Linux. Ke správné funkčnosti je zapotřebí následujícího software: překladač gcc, binutils, patch, bzip2, bison, make, gettext, pkg-config, unzip, libz-dev a libc headers. Po úspěšném zkompilování vznikne binární obraz firmware, který lze aplikovat do zařízení série AirMAX.

5.2. Přístup skrze IPv6

Vzhledem k faktu, že originální firmware nemá do jádra zavedenou podporu IPv6, je potřeba tuto podporu přidat. V tomto případě lze zvolit modul z balíčku. Po zkompilování firmware zjistíme, že byla přidána základní podpora IPv6. V tuto chvíli již lze nastavit IPv6 adresy na rozhraní, získat adresy pomocí Router Advertisement paketů nebo nastavit směrování.

V základním nastavení SSH serveru je povolen přístup z IPv6 adres, není tedy potřeba cokoliv měnit. V případě webserveru, který obstarává program lighttpd však tato podpora standardně nastavena není. V tuto chvíli je potřeba vyhledat konfigurační soubor programu a přidat volbu pro podporu IPv6 (*server.use-ipv6 = "enable"*). Poté můžeme firmware znovu zkompilovat a otestovat podporu.

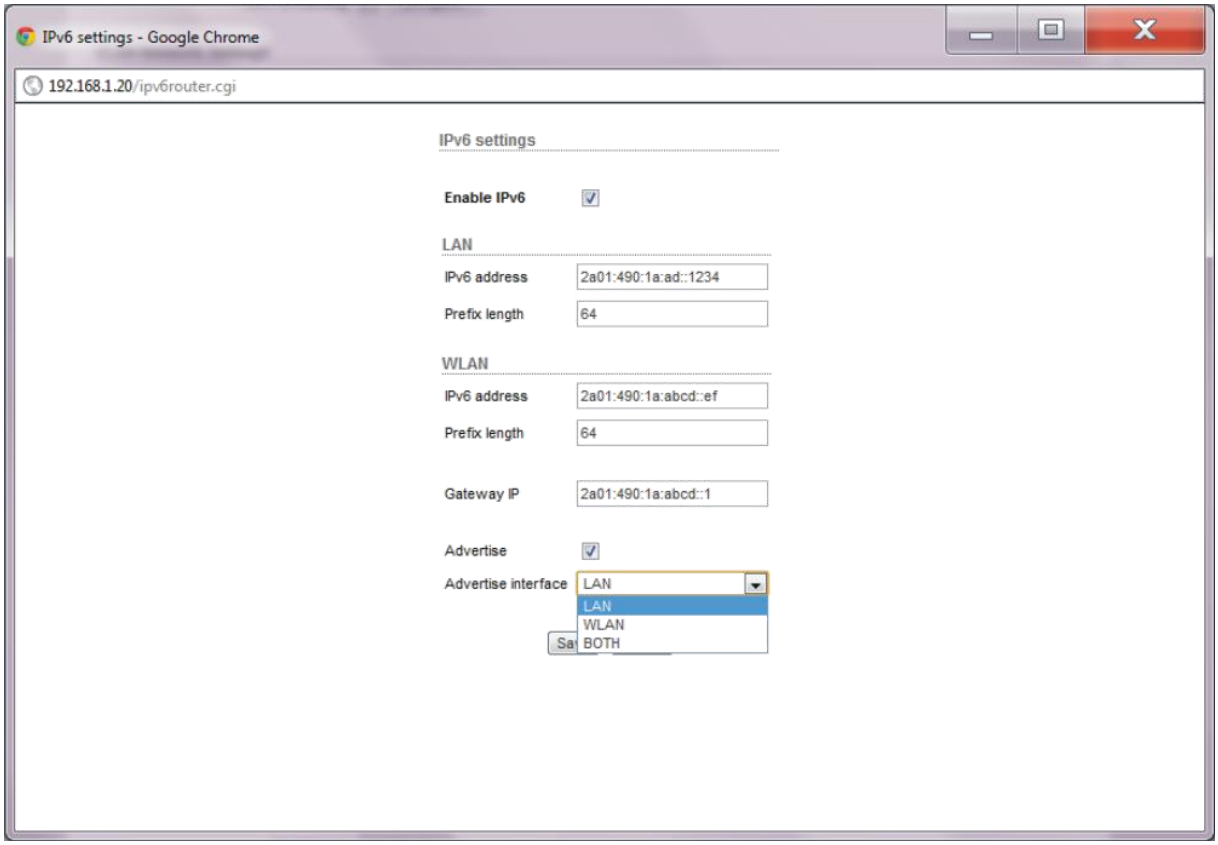
V této chvíli již můžeme začít s úpravou webového rozhraní. Nejdříve je potřeba zajistit rozvržení konfigurace a konfiguračního souboru. Pravděpodobně lze využít originálního konfiguračního souboru a přidání vlastních voleb, avšak vzhledem k tomu, že programy zpracovávající tento soubor jsou distribuovány s uzavřeným zdrojovým kódem, byla zvolena možnost vlastního konfiguračního souboru, aby nedošlo k případné ztrátě či nekonzistenci dat.



Obr. 5.2.1 Nastavení adres v režimu bridge

Konfigurační soubor by měl být, stejně jako v originálním firmware, souborem textovým. Pro čtení a zápis do tohoto souboru tak bylo nutné napsat vlastní obslužnou funkci, která je dále používána při zpracování dat zadaných do webového rozhraní. Pro načtení dat do pole nám bude sloužit funkce *get_ipv6_conf* a pro zápis vícerozměrného pole s parametry poté funkce *write_ipv6_conf*. Ty se nalézají v souboru */lib/ipv6conf.inc* webového adresáře. Dále

zde využijeme část funkcí pro zpracování dat z originálního firmware. Kvůli špatné dostupnosti PHP verze 2, která je z důvodu minimálních nároků na místo a paměť použita v originálním firmware, bylo nutné veškerý vývoj webového rozhraní uskutečnit přímo na jednom ze zařízení. K této verzi PHP je udržena původní dokumentace, která však mnohdy ne zcela přesně koresponduje s danou funkcí. Často tak bylo nutné studovat dokumentaci novější a snažit se upravit výsledný kód tak, aby splňoval syntaxi PHP verze 2. To se v některých případech projevovalo tak, že v novější verzi jsou parametry funkce nepovinné, avšak ve verzi 2 povinné.



Obr. 5.2.2 Nastavení adres v režimu router

Pro přístup po IPv6 je tedy nejprve potřeba nastavit požadovanou adresu. V tuto chvíli je tedy možné na zařízení přistupovat pouze po lokální adrese. Jelikož může být zařízení nastaveno v režimu bridge nebo v režimu router, je potřeba zajistit, aby při konkrétním módu zařízení byla nastavována adresa pro tento režim. Toho lze dosáhnout pomocí již hotové funkce originálního webového rozhraní.

Pro režim bridge definujeme volby povolení IPv6, nastavení adresy autokonfigurací nebo ručně, délku prefixu a výchozí bránu, jak je zobrazeno na obrázku 5.2.1. Veškeré vstupní údaje jsou kontrolovány pomocí funkcí ve skriptovacím jazyku JavaScript. Zde opět využijeme

původní funkce pro zobrazování chybových zpráv a v případě nesprávných vstupních údajů uživatele upozorníme stejně jako v originálním firmware. Příklad chybové zprávy je zobrazen na obrázku 5.2.1.

Obdobně jako pro režim bridge je potřeba vytvořit konfigurační rozhraní pro režim router. Zde musíme nastavit IPv6 adresy pro obě rozhraní a v případě potřeby umožnit propagování prefixu některého z rozhraní bezstavovou automatickou konfigurací. Ukázka konfiguračního rozhraní je zobrazena na obrázku 5.2.2.

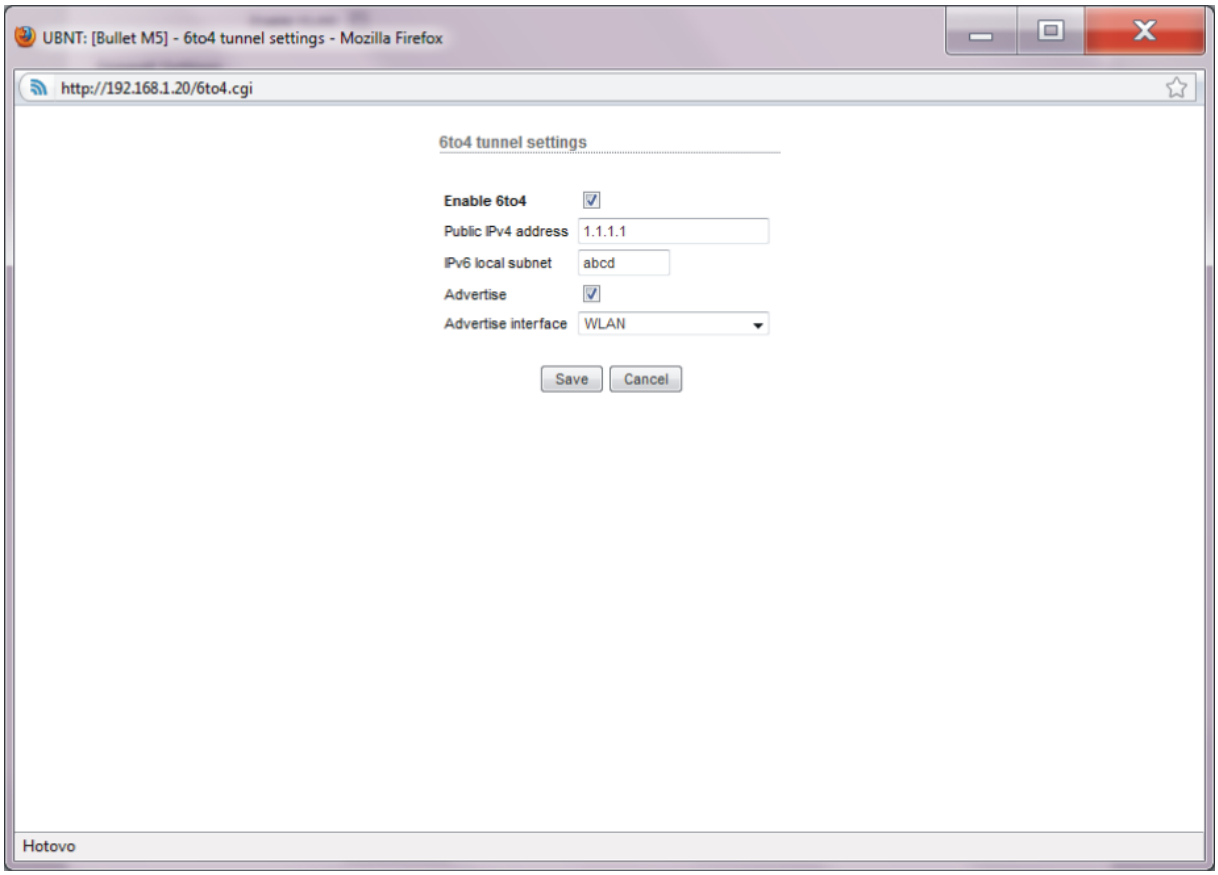
Samotnou aplikaci provedených změn provede obslužný skript, který požadované změny aplikuje. Tento proces je popsán v kapitole 5.7.

5.3. Implementace 6to4

Pro implementaci vytváření 6to4 tunelů je potřeba stejnojmenný softwarový balíček. V balíčkovacím systému AirOS respektive OpenWRT se tento software nachází. Naneštěstí však při kompilaci výsledného image při použití tohoto balíčku dojde k chybě, kdy není nalezen modul *tunnel4.ko*, a proto není možné tento zdroj využít.

Bohužel ani vývojářská komunita OpenWRT, ani vývojáři firmy Ubiquiti nebyli schopni při řešení tohoto problému pomoci, ačkoliv byl tento problém hlášen již vícekrát. Bylo tedy nutné požadovaný balíček zkompilovat ve funkčním systému OpenWRT a stejné hardwarové platformě a chybějící moduly zkopírovat. Tato možnost se po dlouhém řešení problémů jinými – nefunkčními – metodami osvědčila jako funkční. Dále již tedy nic nebránilo vývoji webového rozhraní pro konfiguraci tunelu.

Pro úspěšné vytvoření tunelu je potřeba zadat pouze jednu veřejnou IPv4 adresu. Protože je možné, že směrovač bude disponovat i nativní IPv6 konektivitou, pro případnou delegaci prefixu bezstavovou automatickou konfigurací bylo vhodné přidat i tuto možnost. Jak je patrné na obrázku 5.3.1, bylo přidáno pole pro IPv6 podsíť, možnost delegace prefixu a výběr daného rozhraní. Tato možnost nastaví na zvolené rozhraní IPv6 adresu z prefixu získaného 6to4 tunelem a v případě zapnuté propagace pak tento prefix deleguje automatickou bezstavovou konfigurací dále do sítě.



Obr. 5.3.1 Nastavení 6to4tunelu

5.4. Automatická bezstavová konfigurace

Automatická bezstavová konfigurace je v zařízení použita jak na straně směrovače, který má zasílat ohlašovací zprávy, tak v režimu bridge jako host, který zprávy přijímá a podle nich nastavuje své parametry. Přijímání zpráv a nastavení síťových parametrů je již obsaženo v základní implementaci IPv6, není proto potřeba přidávat jakýkoliv další software. Volbu pro autokonfiguraci tedy můžeme zapnout nebo vypnout příkazem:

```
sysctl -w net.ipv6.conf.br0.accept_ra=1
```

Parametr 0 zde znamená vypnuto a 1 zapnuto.

Pro delegaci prefixů pomocí Router Advertisement zpráv je potřeba nainstalovat další software. Pro zvolené zařízení byl vybrán démon radvd neboli Router Advertisement Daemon. Možností výběru software je samozřejmě více. Delegaci prefixů dokáže zajistit

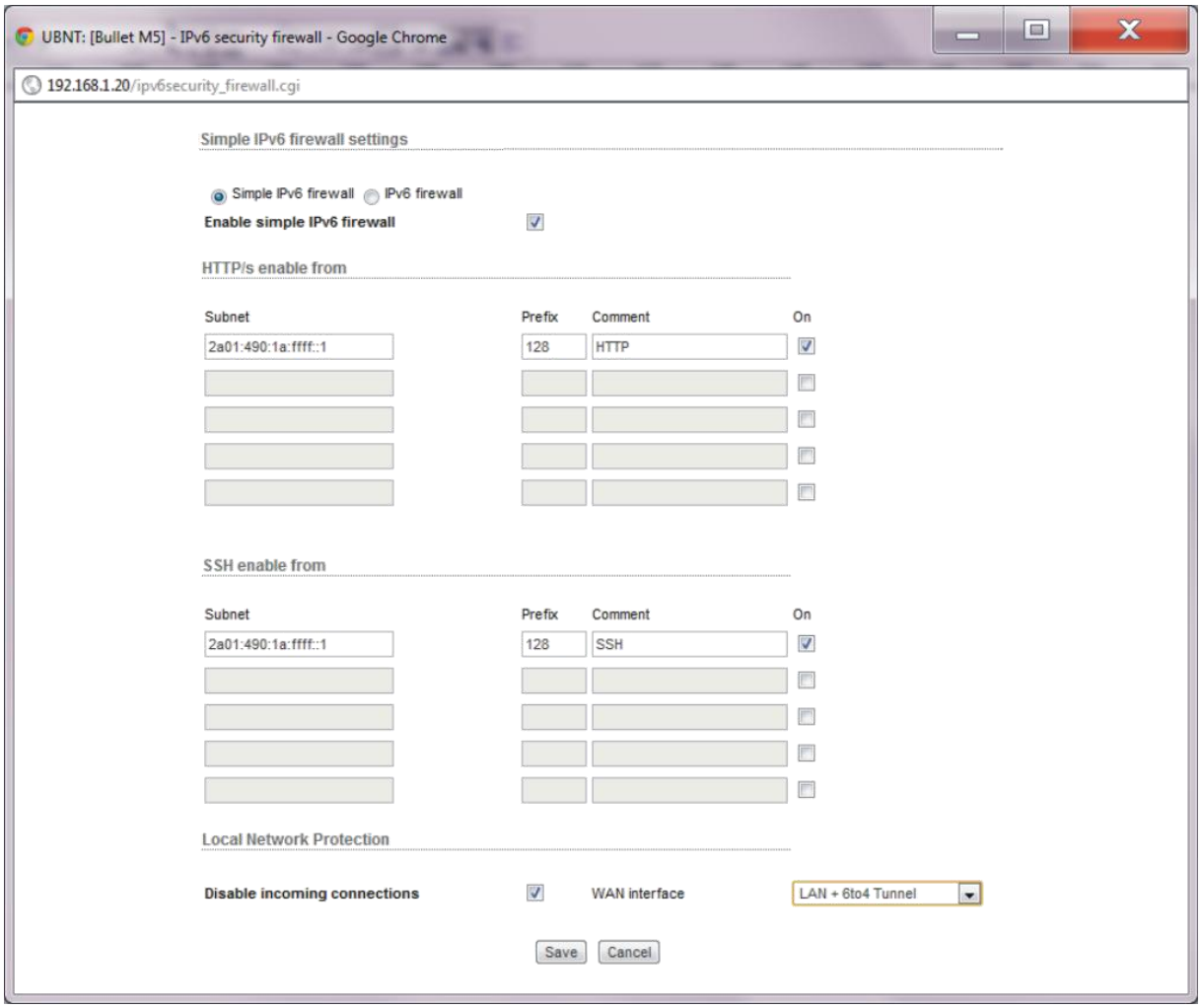
taktéž démon zebra z balíku softwarového směrovače Quagga. Vzhledem k požadavku na minimalizaci použitého místa a paměti byl však zvolen právě démon radvd, který je taktéž zahrnut v balíčkovacím systému. Tento program na základě konfiguračního souboru propaguje do sítě požadovaný prefix. Jelikož směrovač může disponovat jak nativní IPv6 konektivitou, tak i 6to4 tunelem, je potřeba vzít v potaz, že bude zapotřebí delegovat více prefixů z jednoho rozhraní. V manuálových stránkách je uváděno, že pro nastavení rozhraní by měla postačovat volba *prefix ::/64*, která podle IPv6 adres přiřazených k danému rozhraní začne tyto prefixy delegovat. Bohužel tato volba funguje pouze pro první nastavený prefix a pro další již ne. V konfiguračním souboru *radvd.conf* tedy musí být implicitně vypsány oba prefixy.

5.5. Firewall

Při volbě firewallu byla jednou z podmínek využití takzvaného stavového firewallu. Jelikož je v originálním firmware využit program iptables, jako vhodná alternativa pro IPv6 se tedy přímo nabízí ip6tables. Iptables slouží obecně k manipulaci s tabulkami, které využívá netfilter, což je paketový filtr zabudovaný v jádře systému. Pomocí zadaných pravidel ovlivňuje průchod paketů jádrem systému.

V práci jsou zapracovány dva druhy firewallu. Jeden pro řízení přístupu na zařízení a další pak pro pakety, které směrovačem procházejí. Pro řízení přístupu ke konfiguraci zařízení budeme využívat řetězce INPUT, který na vstupu paketu do směrovače zkontroluje, zdali zdrojová adresa na požadovaném portu pochází z povolené podsítě. Paket takto prochází jednotlivými pravidly, dokud některému z nich nevyhoví. V případě, že se tak stane, dále již filtrem neprochází. Z tohoto důvodu je proto potřeba na konci pravidel v řetězci INPUT zakázat veškerý příchozí provoz na požadovaném portu. Pro ilustraci může posloužit následující příklad, kdy je přístup na port 80 webového rozhraní povolen pouze z adresy 2a01:490:1a:ffff::1.

```
p6tables -A INPUT -s 2a0a:490:a1:ffff::1/128 -p tcp --dport 80 -j ACCEPT
ip6tables -A INPUT -s ::/0 -p tcp --dport 80 -j DROP
```



Obr. 5.5.1 Nastavení základního firewallu

V případě, že paket z této adresy pochází, je přijat a další pravidlo se ho již netýká. V opačném případě postoupí na pravidlo následující, kterému však vyhoví každý paket a tento je následně zahozen. Z jakékoliv jiné adresy tedy není možné spojení navázat.

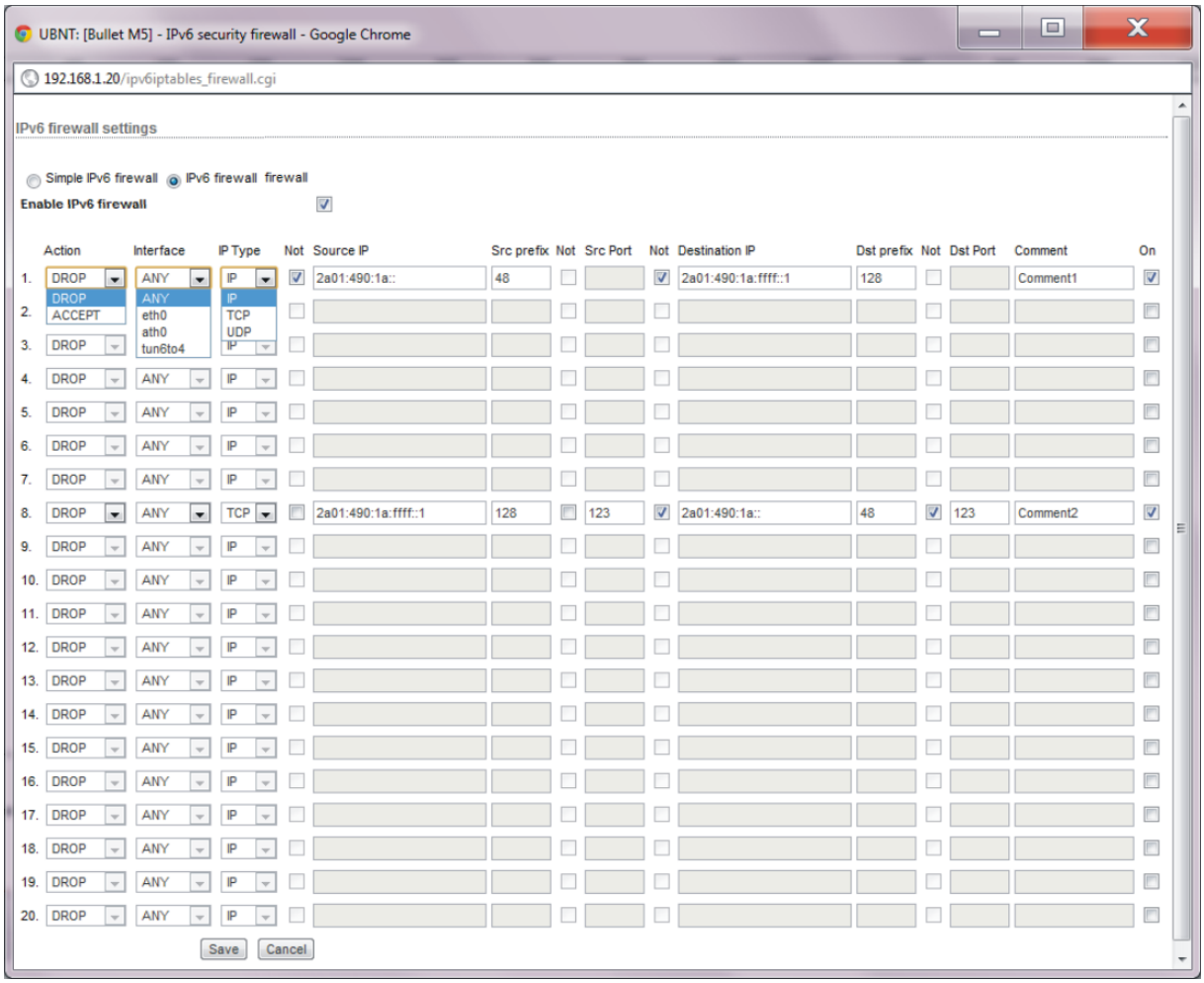
Dalším základním kamenem zabezpečení je zákaz navázání spojení směrem do vnitřní sítě. Zde již budeme vkládat pravidla do řetězce FORWARD. V tomto momentu chceme, aby každé nové spojení, které nebylo navázáno z vnitřní sítě, směrovač dále nepropustil. U tcp toho lze dosáhnout následujícím pravidlem:

```
ip6tables -A FORWARD -i eth0 -p tcp --syn -j DROP
```

Toto pravidlo zahodí každý úvodní paket TCP spojení, který jako vstupní rozhraní využije ethernet (eth0). V případě, že bychom však chtěli využít rozšířeného firewallu, kde bychom si chtěli nadefinovat výjimky, je potřeba toto pravidlo umístit až na konec v řetězci FORWARD.

Tento základní firewall můžeme nastavit z webového rozhraní, které je zobrazeno na obrázku 5.5.1.

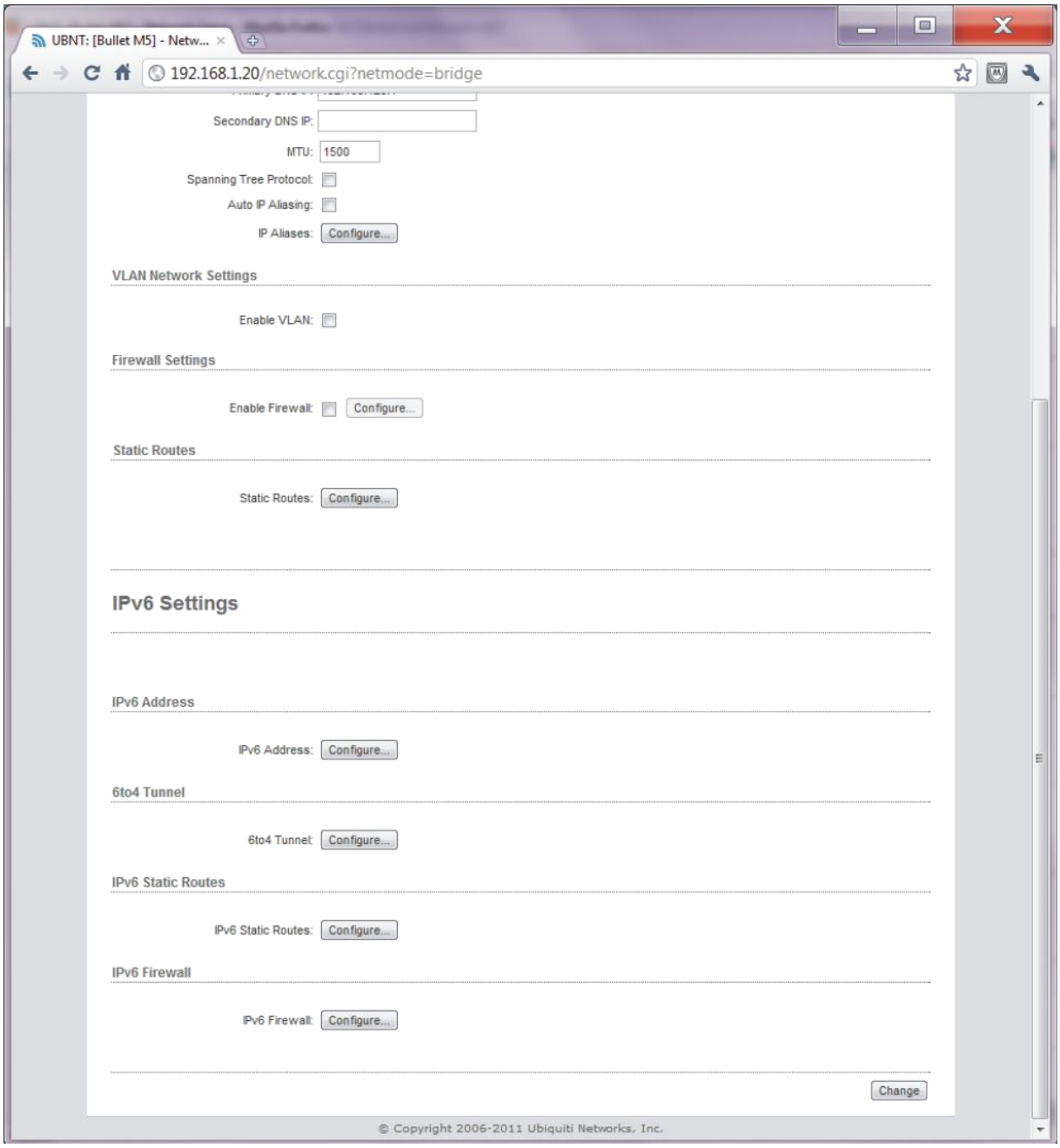
Základní firewall však často nemusí postačovat a uživatelé mohou chtít definovat svůj vlastní firewall, nebo přidat výjimky pro určitá zařízení. K tomuto účelu slouží rozšířená verze IPv6 firewallu. Ta může koexistovat se základní verzí včetně ochrany před navázáním spojení směrem do vnitřní sítě. Rozhraní pro tato nastavení bylo do značné míry převzato z originálního firmware a upraveno pro potřeby IPv6. Rozhraní pro IPv6 firewall je zobrazeno na obrázku 5.5.2 a je zde patrná podobnost právě s originálním rozhraním nastavení firewallu pro IPv4, které je zobrazeno na obrázku 4.1.1.



Obr. 5.5.2 Nastavení rozšířeného firewallu

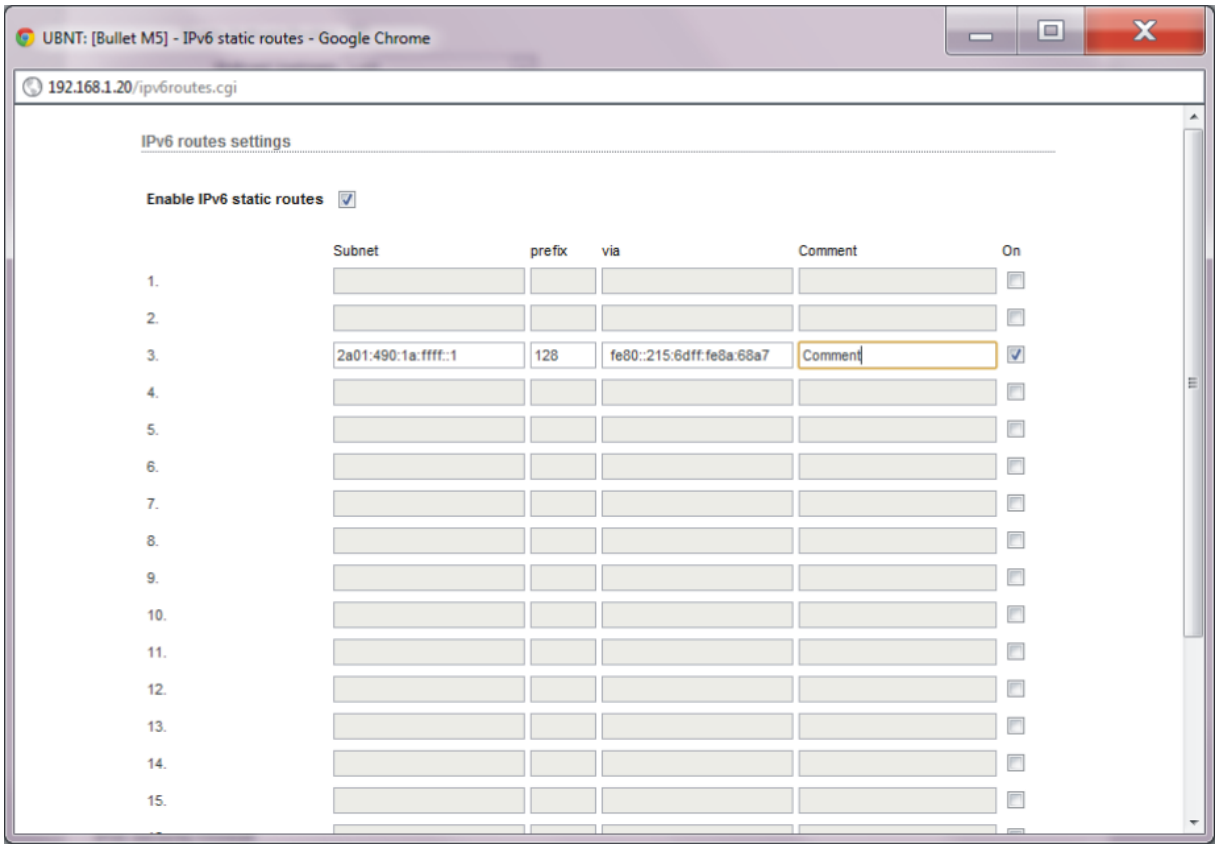
5.6. Ostatní nástroje webového rozhraní

Ke všem výše zmiňovaným volbám, které se týkají síťové konfigurace, lze přistupovat z hlavní záložky *NETWORK* webového konfiguračního rozhraní zařízení, jak je znázorněno na obrázku 5.6.1. Pod nastavení IPv4 byla přidána tlačítka pro nastavení požadovaných parametrů IPv6. Tento způsob byl převzat z originálního firmware, aby byl pro stávající uživatele co nejintuitivnější, jelikož například nastavení firewallu pro IPv4 je zde řešeno obdobnou cestou.



Obr. 5.6.1 Síťové nastavení

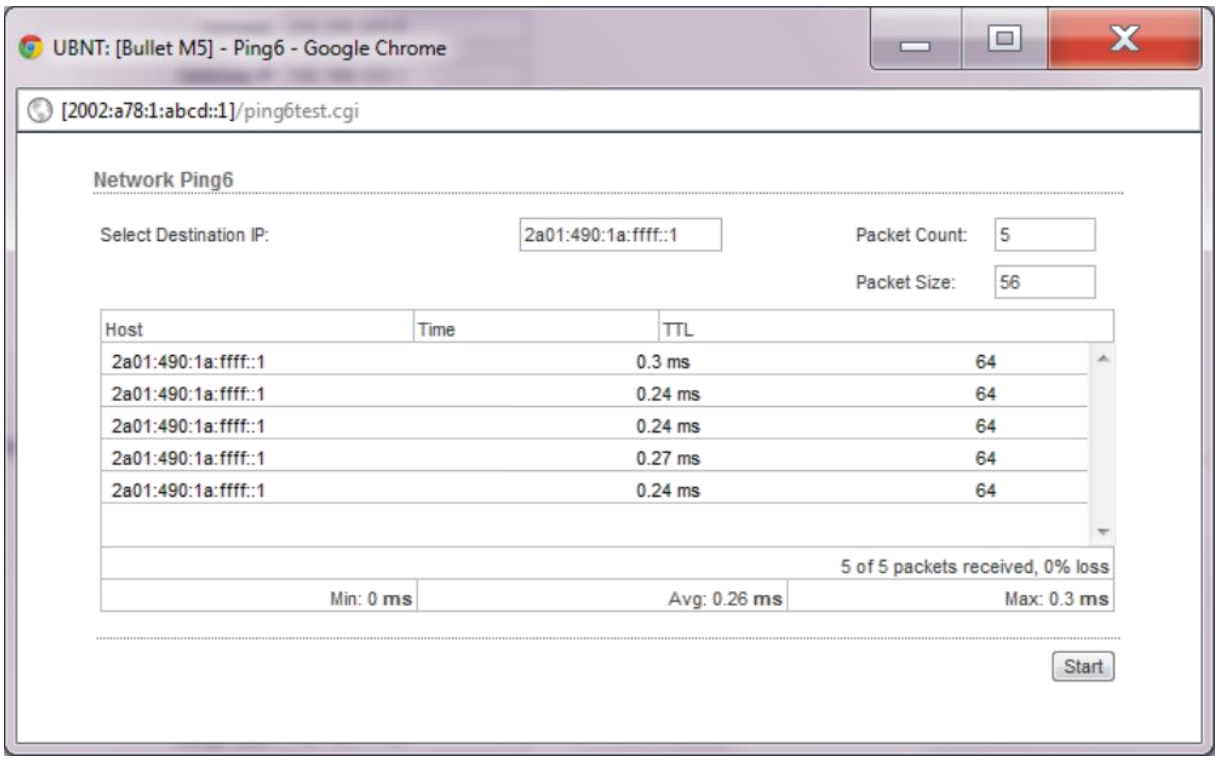
Poslední nastavení, které ještě nebylo zmiňováno a které bylo implementováno do webového rozhraní, je nastavení statického směrování, které je zobrazeno na obrázku 5.6.2. Toto rozhraní bylo opět odvozeno od originálního firmware pro nastavení statického směrování IPv4. Můžeme zde zadat podsíť, velikost podsítě a bránu, kterou budou data do dané sítě odesílána. Každý takovýto záznam můžeme opatřit komentářem a jednotlivé položky povolovat nebo zakazovat. V nastavení pak jednotlivé položky přidáme do zvláštní směrovací tabulky, abychom tak mohli jednoduše spravovat všechny statické záznamy přidané z webového rozhraní, popřípadě tuto tabulku vyprázdnit tak, aby se změna nedotkla případných ostatních záznamů.



Obr. 5.6.2 Nastavení statického routování

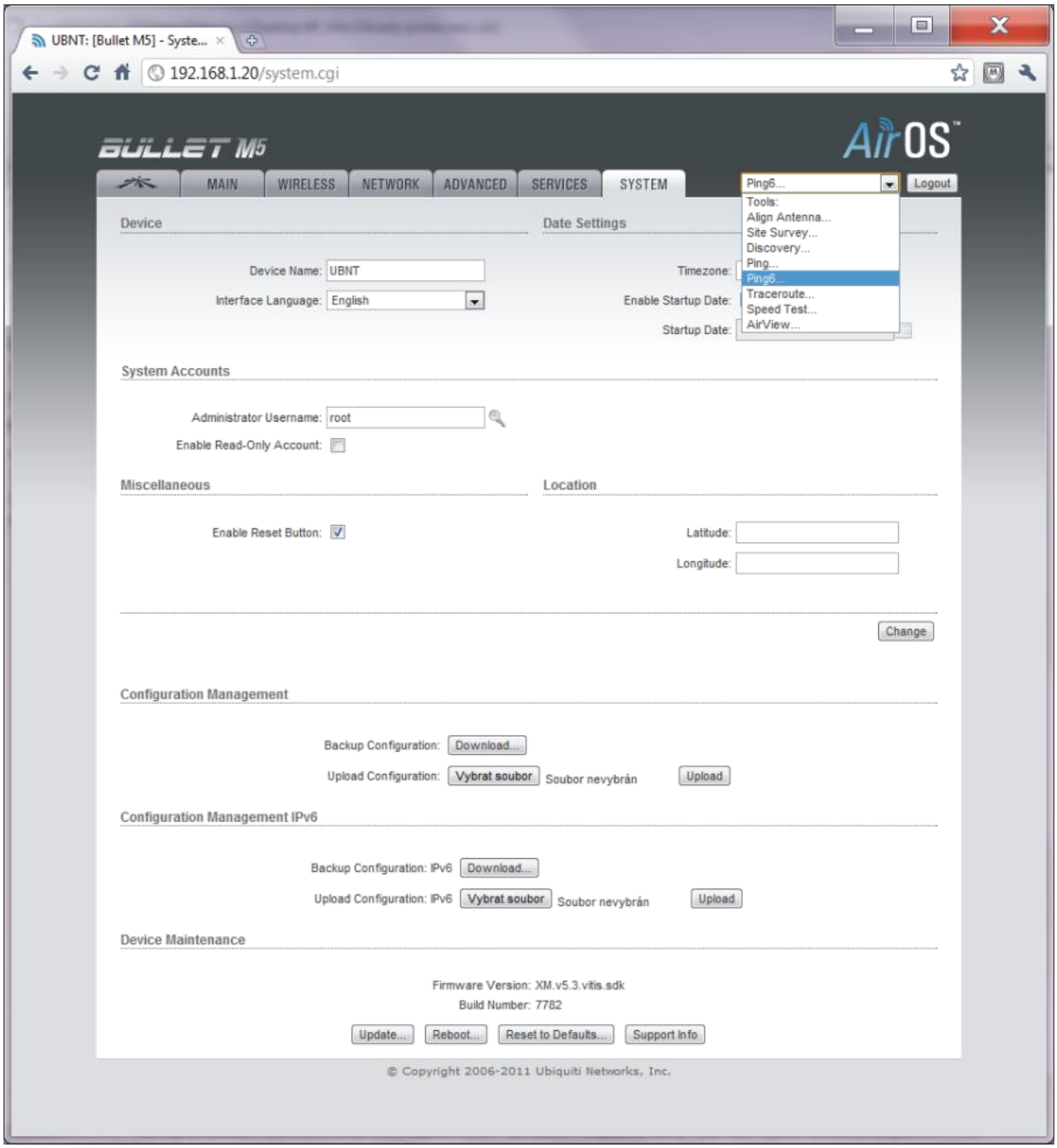
Pro diagnostiku dostupnosti jednotlivých uzlů IPv6 sítě byl přidán nástroj ping6. Webové grafické rozhraní v tomto případě opět vychází z originálního firmware, kde je stejný nástroj implementován pro IPv4. Originální skript vychází z proprietárního obslužného programu pojmenovaného webping firmy Ubiquiti Networks, který je distribuován s uzavřeným zdrojovým kódem. Pro potřeby IPv6 pingu bylo tedy zapotřebí napsat krátký skript, který nahrazuje originální webping pro potřeby IPv6. V této fázi bylo důležité otestovat veškeré

stavy původního skriptu a odezvu na ně ve webovém rozhraní. Výsledek je pak patrný na obrázku 5.6.3.



Obr. 5.6.3 Nástroj pro zjištění dostupnosti uzlů v síti ping6

Poslední úpravou bylo přidání správy nastavení. To bylo přidáno do záložky *SYSTEM*. Zde je možné stáhnout si textový soubor s aktuální konfigurací IPv6 požadovaného zařízení. Je také možné toto nastavení ze souboru obnovit. Při implementaci tohoto nástroje bylo opět vycházeno z originálního firmware, aby bylo ovládání jednotné a pro uživatele co nejintuitivnější. Výsledná implementace i porovnání s originální správou nastavení jsou patrné na obrázku 5.6.4.



Obr. 5.6.4 Správa konfigurace

5.7. Aplikace zvolené konfigurace

Pro aplikaci zvolené konfigurace bylo potřeba vycházet z dostupných softwarových prostředků. Vzhledem ke snížení velikosti volného místa - jak flash paměti, tak i RAM - doinstalováním potřebného software, nebylo možné využít sofistikovanější skriptovací jazyk. Proto se z tohoto pohledu jevilo jako optimální využít skriptování přímo v shellu. Jako

interpret byl tedy použit */bin/sh*, který obsahuje standardní verze AirOS. Pro většinu operací tak stačilo využít některý z již nainstalovaných programů.

Jediný dodatečný program, který bylo potřeba doinstalovat, je obslužný program sipcalc, který slouží jako IP kalkulátor. Tímto programem jsou dopočítávány prefixy požadované velikosti pro záznamy do konfiguračního souboru *radvd.conf* nebo například pro statické routování.

Při aktivaci aplikačního skriptu se původní nastavení smaže a nahraje se nové, aby nevznikaly duplicitní záznamy. Toto probíhá po aktivaci tlačítka *Apply* v případě provedených změn nebo po startu zařízení. Při startu je však uložené nastavení aplikováno automaticky init skriptem, který dále zkontroluje, zdali je vytvořen konfigurační soubor a v případě, že tento soubor neexistuje, jej vytvoří.

6. Testování

Po dokončení implementace navržených úprav bylo nutné otestovat funkčnost těchto změn. Výsledný firmware byl nahrán do zařízení Bullet M5 a zařízení bylo UTP kabelem připojeno do sítě. První fáze testování probíhala na neveřejné IPv4 adrese za NAT směrovačem. K dispozici byla také nativní IPv6 konektivita. Zařízení bylo nastaveno do módu AP a jako stanice posloužil osobní počítač s operačním systémem Windows7.

Nejdříve byla otestována funkčnost nastavení IPv6 adres v režimu *bridge*. Zde byla úspěšně ověřena funkčnost nastavení při zadání statické IPv6 adresy a výchozí brány. Tato nastavení se v systému aplikovala, což bylo ověřeno pomocí příkazů v příkazové řádce:

```
ip -6 addr show
Ip -6 route show
```

Dále byla vyzkoušena funkčnost připojení na webové rozhraní pomocí protokolu IPv6, což se ukázalo jako funkční. Následně byl proveden test spojení do internetu pomocí programu ping6 ze zařízení s tímto výsledkem:

```
ping6 www.mh2net.cz
PING www.mh2net.cz (2a01:490:1a::3): 56 data bytes
64 bytes from 2a01:490:1a::3: seq=0 ttl=60 time=1.618 ms

--- www.mh2net.cz ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 1.618/1.618/1.618 ms
```

Funkčnost byla ověřena i z osobního počítače bezdrátově připojeného k zařízení s pozitivním výsledkem. Následně byla vyzkoušena možnost autokonfigurace zařízení. V systému bylo opět příkazovou řádkou ověřeno reálné nastavení adresy a výchozí brány a opětovně vyzkoušena funkčnost spojení.

Dále byla otestována tvorba 6to4 tunelu. Pro zařízení byla přiřazena jedna veřejná IPv4 adresa metodou NAT1:1. Bohužel v této konfiguraci se nepodařilo 6to4 tunel vytvořit, jelikož zařízení nemělo tuto veřejnou IPv4 adresu fyzicky nastavenou.

V režimu bridge byl pak otestován základní firewall, kdy byl omezen přístup pouze na IPv6 adresu osobního počítače a poté úspěšně vyzkoušena funkčnost navázání spojení. Potom byla IPv6 adresa osobního počítače manuálně změněna a znovu vyzkoušen přístup ke konfiguraci zařízení. V této chvíli však nebylo navázáno spojení, což potvrdilo funkčnost nastavení. Pro kontrolu byl firewall na zařízení vypnut a poté již bylo spojení se zařízením navázáno i z manuálně nastavené IPv6 adresy.

Následně bylo zařízení přepnuto do režimu *router* a na výchozím směrovači byl směrován veřejný IPv6 prefix na IPv6 adresu určenou pro toto zařízení. Potom byla manuálně nastavena IPv6 adresa rozhraní LAN a WLAN. Na osobním počítači byla dále manuálně nastavena IPv6 adresa z rozsahu rozhraní WLAN a odzkoušeno spojení jak ze zařízení, tak i z osobního počítače. V obou případech bylo navázání spojení úspěšné a bylo navázáno nativní IPv6 konektivitou.

V dalším kroku bylo zařízení nastaveno tak, aby delegovalo IPv6 prefix na rozhraní WLAN pomocí Router Advertisement. Na osobním počítači bylo následně v příkazové řádce ověřeno nastavení síťového rozhraní pomocí příkazu ipconfig.

```
Adaptér bezdrátové sítě LAN Bezdrátové připojení k síti:

Přípona DNS podle připojení ...:
IPv6 adresa.....: 2a01:490:1a:abcd:5535:f9bb:d3a1:6e6d
Dočasná IPv6 adresa.....: 2a01:490:1a:abcd:85f6:903f:967c:156f
Místní IPv6 adresa v rámci propojení...: fe80::5535:f9bb:d3a1:6e6d%12
Adresa IPv4 . . . . . : 192.168.123.169
Maska podsítě . . . . . : 255.255.255.0
Výchozí brána . . . . . : fe80::215:6dff:fe8a:68a7%12
                          192.168.123.1
```

Pak byla opětovně úspěšně vyzkoušena funkčnost připojení pomocí příkazu ping na osobním počítači.

Následovalo otestování nastavení statického směrování. V tuto chvíli bylo za pomoci přepínače do sítě připojeno další zařízení (směrovač), které mělo na jednom rozhraní nastavenou IPv6 adresu ze stejného rozsahu jako testované zařízení na rozhraní LAN a na druhém rozhraní pak nastavenou adresu z jiného rozsahu. V nastavení směrování pak byla tato podsíť směrována přímo na adresu přidaného směrovače. Pomocí programu *tcpdump* a *traceroute* byla pozitivně ověřena funkčnost požadovaného nastavení.

V režimu *router* bylo také možné otestovat rozšířený firewall a zákaz navázání příchozích spojení. Nejprve byl tedy otestován zákaz navázání příchozího spojení. Na osobním počítači byl nainstalován webový server *Apache2* a ze zařízení umístěného v internetu úspěšně vyzkoušeno navázání spojení. Následně byla nastavena volba pro zakázání příchozího spojení a jako WAN vybráno rozhraní LAN. Pak bylo znovu ověřeno navázání spojení avšak tentokrát s negativním výsledkem. Pro ověření funkčnosti navázání spojení pouze z vnitřní sítě tedy bylo potřeba otestovat navázání spojení v opačném směru, což se podařilo.

Pro otestování rozšířeného firewallu byl na osobním počítači nainstalován program *CesarFTP* pro tvorbu ftp serveru. V tuto chvíli tedy na osobním počítači pracoval webový server na portu 80 a ftp server na portu 21. Pro ověření funkčnosti byl firewall vypnut a úspěšně vyzkoušeno navázání spojení. Poté byl zapnut zákaz navázání příchozích spojení a ověřeno, že spojení opravdu nelze navázat. Pro otestování rozšířeného firewallu bylo nastaveno pravidlo, které na portu 80 IPv6 adresy osobního počítače tyto pakety akceptuje. V následujícím testu se tedy povedlo z internetu navázat spojení pouze na portu 80 webového serveru. Spojení na ftp server se navázat nepodařilo.

Poslední neotestovanou částí se tak stalo vytvoření 6to4 tunelu. Bylo tedy potřeba využít nativní veřejné IPv4 adresy. V této fázi testování tedy bylo nutné zajistit takovouto IPv4 adresu, a proto další test probíhal v serverovně o.s. mh2net. Zařízení bylo zapojeno totožně jako v prvním testu.

V tuto chvíli byl nastaven režim *router* a nebyla povolena nativní IPv6 konektivita. Na rozhraní LAN byla nastavena veřejná IPv4 adresa 81.201.61.79, která byla taktéž vyplněna v konfiguraci 6to4 tunelu. Prefix byl nastaven na hodnotu *abcd* a byla zapnuta propagace tohoto prefixu do rozhraní WLAN. V nastavení osobního počítače pak bylo zkontrolováno, že je zde přidělena IPv6 adresa z požadovaného rozsahu. Pro testování spojení posloužil webový server umístěný v serverovně, který disponoval nativní IPv4 i IPv6 konektivitou.

Funkčnost 6to4 tunelu tak mohla být jednoduše ověřena i pomocí programu *ping* a *ping6*, kdy je patrný rozdíl latencí, což způsobila komunikace se zprostředkovatelem 6to4 tunelu.

```
ping6 www.mh2net.cz
PING www.mh2net.cz (2a01:490:1a::3): 56 data bytes
64 bytes from 2a01:490:1a::3: seq=0 ttl=60 time=4.889 ms

--- www.mh2net.cz ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 4.889/4.889/4.889 ms

 ping www.mh2net.cz
PING www.mh2net.cz (81.201.61.3): 56 data bytes
64 bytes from 81.201.61.3: seq=0 ttl=63 time=0.826 ms

--- www.mh2net.cz ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
    round-trip min/avg/max = 0.826/0.826/0.826 ms
```

Poté byly také analyzovány IPv4 pakety putující ze zařízení ke zprostředkovateli, ze kterých je patrné zapouzdření IPv6 datagramů do IPv4.

```
IP a14.mh2net.cz > 192.88.99.1: IP6 2002:51c9:3d4f::1 > 2a01:490:1a::3:
ICMP6, echo request, seq 2, length 64
IP 192.88.99.1 > a14.mh2net.cz: IP6 2a01:490:1a::3 > 2002:51c9:3d4f::1:
ICMP6, echo reply, seq 2, length 64
```

Následně byla testována kombinace koexistence nativní IPv6 konektivity a 6to4 tunelu. V této kombinaci byla preferována nativní konektivita, avšak nebylo možné nasimulovat chování systému při výpadku nativní konektivity a zachování pouze 6to4 tunelu. Dále bylo nastavení 6to4 tunelu testováno ještě v režimu *bridge*, které bylo taktéž funkční.

7. Závěr

Úprava firmware založeného na operačním systému OpenWRT pro zařízení firmy Ubiquiti Networks, Inc., která by umožnila zavedení IPv6, se ukazuje jako možná. V průběhu řešení bakalářské práce se podařilo navrhnout a implementovat podporu IPv6 a požadované úpravy do domácích směrovačů této firmy. Implementace samotná pak zahrnuje odpovídající prvky grafického webového rozhraní, přes které je možné všechny potřebné prvky ovládat. Při praktickém řešení nastalo několik zásadních problémů, které se však podařilo vyřešit.

Na funkčním vzorku se následně podařilo otestovat všechny požadované funkce v nejrůznějších kombinacích, které by mohly být v praxi nastaveny. Nebylo však možné otestovat všechny možné kombinace v reálném provozu, a proto je možné, že při specifických podmínkách by se mohl objevit neočekávaný problém. Vzhledem k tomu, že je řešení IPv6 na tomto zařízení konfiguračně odděleno od originálního firmware, případná chyba by tedy neovlivnila správné fungování originálního nastavení.

Jako další možné rozšíření úprav by mohlo být přidání podpory pro DHCPv6, kdy by směrovač dostal přidělen vlastní prefix a ten by následně propagoval do místní sítě. Dále by se mohlo zdokonalit webové rozhraní pro firewall tak, aby bylo možné tímto rozhraním kompletně nahradit příkazový řádek.

Za dobu tvorby této bakalářské práce byly postupně uvolněny tři oficiální verze firmware. Jelikož je takováto úprava možná, lze předpokládat, že postupem času bude implementována podpora IPv6 i do originálního firmware.

Seznámení se s vývojovým prostředím pro úpravu firmware pro domácí zařízení a prohloubení znalostí z oblasti IPv6 pokládám za osobní přínos této práce a věřím, že takto získané zkušenosti zúročím i v budoucnu.

Seznam použité literatury

[1] International Telecomunication Union. *ITU* [online]. 2011, 2011-03-04 [cit. 2011-03-04]. Dostupné z WWW: <http://www.itu.int/>.

[2] Internet Assigned Numbers Authority (IANA). *IANA - Internet Assigned Numbers Authority* [online]. 2011 [cit. 2011-04-01]. Dostupné z WWW: <http://www.iana.org/>.

[3] The Internet Engineering Task Force (IETF). *The Internet Engineering Task Force (IETF)* [online]. 2011 [cit. 2011-04-01]. Dostupné z WWW: <http://www.ietf.org/>.

[4] SATRAPA, Pavel. IPv6 : internetový protokol IPv6. Praha : CZ.NIC, 2008. 357 s. ISBN 978-80-904248-0-7.

[5] RFC 791. *INTERNET PROTOCOL*. California : Information Sciences Institute University of Southern California, 1981. 45 s.

[6] RFC 1550. *IP: Next Generation (IPng) White Paper Solicitation*. Harvard University : Network Working Group, 1993. 6 s.

[7] RFC 1752. *The Recommendation for the IP Next Generation Protocol*. Harvard University : Network Working Group, 1995. 52 s.

[8] RFC 1883. *Internet Protocol, Version 6 (IPv6) Specification*. [s.l.] : Network Working Group, 1995. 37 s.

[9] RFC 3701. *6bone (IPv6 Testing Address Allocation) Phaseout*. [s.l.] : Network Working Group, 2004. 6 s.

[10] RFC 2460. *Internet Protocol, Version 6 (IPv6) Specification*. [s.l.] : Network Working Group, 1998. 39 s.

[11] RFC 1918. *Internet Protocol, Version 6 (IPv6) Specification*. [s.l.] : Address Allocation for Private Internets, 1996. 9 s.

[12] EU. ADVANCING THE INTERNET. In *Action Plan for the deployment of Internet Protocol version 6 (IPv6) in Europe*. 27/05/2008.

[13] Usnesení vlády č. 727/2009 Sb. ke Zprávě o přechodu na internetový protokol verze 6 (IPv6).

[14] Free Software Foundation, Inc. *Licenses - GNU Project - Free Software Foundation* [online]. 2010/04/27 [cit. 2011-04-01]. Dostupné z WWW: <http://www.gnu.org/licenses/>.

[15] APNIC. *APNIC IPv4 Address Pool Reaches Final /8* [online]. 2011-04-15 [cit. 2011-04-16]. APNIC. Dostupné z WWW: <http://www.apnic.net/publications/news/2011/final-8>.

[16] RFC 1971. *IPv6 Stateless Address Autoconfiguration*. [s.l.] : Network Working Group , 1996. 23 s.

[17] RFC 5006. *IPv6 Router Advertisement Option for DNS Configuration*. [s.l.] : Network Working Group, 2007. 12 s.

[18] RFC 6106. *IPv6 Router Advertisement Options for DNS Configuration*. [s.l.] : Internet Engineering Task Force (IETF), 2010. 19 s.

[19] RFC 3056. *Connection of IPv6 Domains via IPv4 Clouds*. [s.l.] : Network Working Group , 2001. 23 s.

[20] RFC 6092. *Recommended Simple Security Capabilities inCustomer Premises Equipment (CPE) forProviding Residential IPv6 Internet Service*. [s.l.] : Internet Engineering Task Force (IETF) , 2011. 36 s.